Royal Holloway, University of London

Doctoral Thesis

Virtual Exploration of the Human Vocal Tract

Author:

Daniel RM Woods

Supervisor:

Professor David M Howard

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Biosignals and Intelligent Systems Group
Electronic Engineering

December 2023

Abstract

Hypothesis: By simulating the acoustic field throughout the entire vocal tract the evolution of speech sounds within the tract can be directly and quantitatively related to physical variations in the tract geometry. This insight into speech production could then be applied to a variety of fields where the ability to alter or investigate speech characteristics in a targeted way could be useful for example in the teaching of speech science, in speech coaching, or as part of the planning of medical procedures. In this research, a bespoke acoustic simulation package has been produced using a continuous 3-dimensional Digital Waveguide Mesh (DWM) which can produce acoustic output throughout the entire simulation domain containing the tract at every time step. This package has been shown to reproduce formant frequencies for a variety of vocal tract shapes with an average mean absolute error of 10.12 % at the lips, which is comparable to other research. These results have been investigated by comparing simulation output to recorded output from physical models. This simulation package has also been used to perform studies into the shifting of formant frequencies during speech sound production along the length of the tract, and into the effect on formant frequencies of the removal of geometric features of the tract such as the piriform fossae. These studies have been compared to physical internal measurements of vocal tract models from living subjects, showing preliminary agreement with further development required. A large emphasis has been placed on the accessibility of this research, with the production of several tools for visualisation of the data contained within, and with decisions made during the production of the simulation package itself.

Declaration of Authorship

I hereby declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this or any other University. All sources are explicitly stated and referenced. I also declare that some parts of the work in this thesis have been presented previously, at conferences and in journals, in the following publications:

- **Virtual Exploration of The Human Vocal Tract**, D. R. M. Woods, presented at the 50th Voice Foundation Annual Symposium (Care of the Professional Voice), virtually attended, 2nd June - 6th June 2021 [1].

- **Using Voice Synthesis Techniques to Virtually Explore the Sound Field within the Human Vocal Tract**, D. R. M. Woods and D. M. Howard, presented at the 51st Voice Foundation Annual Symposium (Care of the Professional Voice), Philadelphia, USA, 1st June - 5th June 2022 [2].

Acknowledgements

My journey through the PhD has been a long and circuitous one. Beginning in Physics and ending in Electronic Engineering, persisting through a global pandemic, house moves, and many other life-defining events. To say that I am a different person to the one that began this process would be a gross understatement. Doing a PhD is at times all consuming, and persevering to the other side is only possible with enough people around you that really do care about your well-being and success.

I would first like to thank the technical team in the RHUL Electronic Engineering department, especially Alex and Dave. My visits to the office only made up a small part of your week and were few and far between, but the technical support provided to me was key to this work and it would not be nearly as complete without your patience and assistance.

I have had the opportunity to discover a passion and aptitude for teaching during my studies and would like to thank everyone that taught alongside me or gave me the opportunity to teach both in the Physics and EE department. Weekly teaching has been a respite for me from the solitary nature of doing a PhD, and these sessions allowed me to socialise with other students who knew what I was going through. Thank you to Ian, Asher, Stef, Adriana, Jacob and Andrew for teaching with me, teaching me, and being there for a chat when things were quiet.

My friends, both the ones that have been with me before this process and the ones made along the way, have kept me sane through this whole process and I could not have done this without them. They have made me laugh, made me cry, made me furious, and mostly just made me stay human. There are too many to mention but I would specifically like to thank Siobhan, Liam, Emma, Rory, Andrew, Jacob, Jack, Yuji, Ella, Jaya, Ben, Georgia, Tana, Charlie, Kamil, and probably more that I am forgetting.

I have had the extreme luck of having an excellent supervisor during this PhD, who has supported me through the difficult and unclear path that so many PhDs seem to tread. I often came in to meetings feeling stressed, feeling behind on work,

and feeling like I didn't belong in research. I left every meeting feeling confident, happy, and excited. Thank you to David for being endlessly positive, supportive, and generally excited about what I was doing.

Finally, I really wouldn't still be doing this if it wasn't for my intelligent, beautiful, and generally amazing partner. Floss, this really is entirely dedicated to you. It almost feels unfair to take credit for anything contained within, because I can't even count the number of times I might have given up if you weren't there alongside me going through the same thing and encouraging me along the way. Thank you for being there and thank you for sharing this experience, and hopefully many more, with me.

And to all those who come after: good luck, push on, take some breaks, and don't lose.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Hypothesis and Motivation

The human voice is the framework within which people have interacted with the world around them since the beginning of human society. In a modern society, the voice is still central to the way people experience the world and convey information. The mechanisms of speech production are well understood from a biological and phonetic standpoint, however this information is difficult to access without deep technical knowledge. These mechanisms are also most commonly described by considering the vocal tract as an object which transforms a source signal into an output speech sound which varies based on the articulation of the tract for that sound. This is a valid approach, but does not preserve any direct links between the individual aspects of the articulation of the tract and the output it produces

The hypothesis of this work states that **by simulating the acoustic field throughout the entire vocal tract the evolution of speech sounds within the tract can be directly and quantitatively related to physical variations in the tract geometry**. Through exposing the process of speech production within the tract in terms of visual and intuitive concepts like the shape of the tract and its side passages and constrictions, an audience which lacks a strong background in speech science and acoustics can begin to make informed choices in fields surrounding the voice. There is often a need for professional singers to adjust individual formant

frequencies during speech production to access different ranges of the human voice [12]. By using vocal tract simulations, it would be possible to identify multiple ways of creating a frequency shift in specific parts of the speech spectrum that would be directly suitable to an individual's tract geometry.

People who feel disconnected from the way their voice sounds, for reasons of gender dysphoria or otherwise, frequently undergo speech therapy with the goal of training them to naturally articulate their vocal tract in such a way as to raise the pitch of their voice or attempt to align individual speech formants with target values [13, 14]. Using work built on this research, a person could be given a more targetted plan for making these changes in a way that would require the least change to their specific natural articulation. Further still, it is not uncommon that an individual must resort to surgical reconstruction due to issues mentioned above, prior medical procedures which left the voice permanently changed or inoperable, or even traumatic incidents which led to damage to the tract [15, 16]. Future work in this field could directly inform the surgical plan by providing guidance as to which parts of the reconstruction would lead to the intended speech sound either based on prior recordings of the individuals speech or based on desired speech characteristics.

Much previous research has largely been concerned with the study of acoustic propagation within the vocal tract for the purpose of speech synthesis, which is an important and ever expanding field in a modern digital world driven by human-computer interfacing. Inherently speech synthesis research is often focussed on speech intelligibility and accuracy rather than reproduction of the underlying biological mechanisms. The field of vocal tract acoustic modelling however takes the intuitive approach that accurately modelling speech production would ideally lead to truly accurate speech output which is indistinguishable from that which a human would produce.

This so-called 'physical modelling' approach to speech synthesis is itself a broad field, encompassing electrical analogues with manually tunable parameters to fully programmatic simulations of the flow of acoustic pressure through a simulation

domain by approximately solving the acoustic wave equations [17, 18]. This is a field of compromises. Models based on physical analogues usually operate in real time and are relatively simple to adjust but typically suffer in output accuracy due to the required simplifications in the implementation. Acoustic simulations can theoretically produce exact solutions to the acoustic propagation in the tract, but their accuracy is typically directly proportional to their complexity and the amount of time taken to perform the simulation which limits their use in dynamic real world applications.

While this research is not directly interested in physical modelling for speech synthesis, many of the advancements made in that field make this work possible. This work will present a bespoke modelling package based on a 3-dimensional Digital Waveguide Mesh (DWM) which produces the acoustic response of the human vocal tract continuously throughout the entire simulation domain. The package is designed to be useable without large amounts of technical knowledge by accepting arbitrary geometries and performing all pre-processing and analysis steps automatically, while still producing the raw data for more thorough exploration. This package has been used to investigate the shifting resonant frequencies in the tract along its length to provide an insight into how the geometrical properties of the tract affect sound production continuously. It has also been used to investigate the effect of removing parts of the vocal tract, for example the piriform fossae, on the production of speech while relating the acoustic output back to the geometry.

This research represents an avenue for improved understanding of speech production for a less technical audience by relating acoustic output back to geometry using simple physical concepts. A focus on this approachability is present throughout the whole work, both in the design of the modelling package and in the presentation of the data through visualisations in virtual spaces. It is clear that a further developed version of this research would have practical applications in all the ways previously described. This work is still only a first step on this path to approachable and intuitive understanding and interrogation of speech production, but the potential

applications should label this field as a research priority.

1.2 Layout

Chapter 2 of this thesis provides a theoretical introduction to physical acoustics in the vocal tract and how speech in produced. Once this basis of knowledge has been laid out, a series of methods for simulating the physical acoustics in real environments are presented, with discussion of their particular formulations, advantages, and disadvantages. This chapter also discusses some advances in modern acoustic modelling techniques such as parallelisation and real-time acoustic simulations which have come about due to the rise in popularity of Virtual Reality (VR).

Chapter 3 contains a review of explorations of pre-existing modelling packages that could have been used to perform the acoustic simulations required for this work, discussing their advantages and disadvantages relating to this project.

Having determined that none of the packages explored in this work were suitable for the requirements of this research, **Chapter 4** focuses on the creation and validation of a functional acoustic propagation algorithm based on the Digital Waveguide Mesh (DWM) algorithm within the Python language. Explorations of 3D modelling and voxelisation, as well as of an Finite-Difference Time-Domain (FDTD) method which was briefly considered as the algorithm of choice for this work. This chapter also includes validation of the acoustic propagation algorithm against both formant values stated in literature and against physical recordings of fabricated 3D versions of the simulated vocal tracts, the breadth of which is a novelty of this research.

Chapter 5 presents the complete vocal tract simulation package produced from this work. Details are included of additional development required on top of the previously discussed propagation algorithm. This chapter also includes a presentation of some usages of VTSim to interrogate the acoustic response of the vocal tract, specifically in resonant frequency analysis along the tract and in investigation of how those resonant frequencies shift when the tract geometry is edited.

While the accuracy of outputs within the tract was asserted based on accuracy

of outputs taken at the lips, **Chapter 6** features an experimental validation of internal simulation against measurements made within a 3D printed vocal tract. The method and outcomes of this measurement are described thoroughly within and form another novelty of this work.

Chapter 7 shows some examples of data visualisations designed to provide an intuitive understanding of speech production using simulation data and clear examples. Information on how these visualisations were made is available in this chapter for future works.

Finally, **Chapter 8** provides a summary of the work contained within this thesis, the novel contributions of this work, an analysis of the success of this work in answering its hypothesis, and a discussion of potential future works in this field.

Chapter 2

Acoustics and Computational Modelling

To model the human voice in a way which is physically accurate requires knowledge of both the fundamental physics which governs acoustic propagation and the ways in which those physics can be approximated, simplified, and solved. Sections 2.1 and 2.2 will provide an overview of the minimum rigour required to understand the processes that modern acoustics simulations are built upon, and a basic method of speech synthesis which is commonly used in this field. These sections are largely based on Kinsler's et al. 'Fundamentals of Acoustics' [19], which contains a much more thorough treatment of the concepts than is required here.

The wave equations and their solutions only provide an accurate solution to the acoustic propagation in free space. Environmental acoustics in real spaces with even simple boundary geometries requires significant adjustments and additions to these equations. For models which will be applied to real situations the solutions must take into account the behaviour of the sound field and how it interacts with itself and the surrounding environment, across a large frequency range.

The latter part of this chapter contains a review of a wide variety of acoustic modelling techniques that are or have been used to approximate these complex acoustic environments for computational modelling. Section 2.3 covers modelling techniques

based on the assumptions commonly found in geometrical optics of treating acoustic propagation as a process driven by elastic collisions and reflections of point-, ray-, or volume-like acoustic fronts. Section 2.4 shows a more accurate approach of encompassing the wave-based behaviours of acoustic fields by solving the wave equation in some way. Finally, Section 2.5 contains an overview of some modern advancements in acoustic modelling, including the usage of more sophisticated hardware and advances spurred on by the advent of Virtual Reality (VR) and the need for real-time immersive audio.

2.1 Acoustics of Speech Production

Sound is the result of a pressure wave that travels through a medium. The resultant pressure at a point due to the variation caused by a travelling wave, the acoustic pressure p, can be defined as

$$p(x, y, z) = P(x, y, z) - P_0(x, y, z), \tag{2.1}$$

where P is the instantaneous pressure and P_0 the ambient pressure in the fluid at that point. As the waves travel over the point, a waveform $p(x, y, z)$ is produced which varies with time. Energy loss in the system comes from the characteristic acoustic impedance z which is a measure of the resistance of the medium to the variation in pressure. The form on the acoustic impedance varies depending on conditions and dimensionality. For a planar wave z is defined as

$$z = \rho c, \tag{2.2}$$

where ρ is the density of the medium and c is the speed of sound within the medium. This is related to the acoustic impedance at a given acoustic surface, Z, via

$$Z = z/S = \frac{p}{v}\ , \tag{2.3}$$

where S is the area of the surface, p is the acoustic pressure, and v is the velocity of particles in the acoustic medium. It should be noted that this is strongly reminiscent of Ohm's law within electronics, with p acting as the energy differential which encourages flow and v as the flow of the acoustic 'current'.

Acoustic waves are also described by the wave equations, a set of partial differential equations that govern wave propagation. The simplest wave equation, the 1D case, is as follows:

$$\frac{\partial^2 p}{\partial t^2} = c^2 \frac{\partial^2 p}{\partial x^2}, \tag{2.4}$$

where p is a function of the position x and the time t and c is the speed of propagation, in this case the speed of sound. The standard solution to these equations, as discovered by d'Alembert in 1746, is to use a travelling wave solution which treats p as a linear combination of two wave components travelling in opposite directions. These functions are well-behaved in differentiation and describe p through their summation at any point in the space. The wave equation can be described for an arbitrary number of dimensions using the Laplace operator:

$$\begin{aligned} \frac{\partial^2 p}{\partial t^2} &= c^2 \nabla^2 p \\ \nabla^2 &= \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + ... \end{aligned} \tag{2.5}$$

While these equations fully describe propagation in unbounded free space, for the purposes of considering the acoustics of the human vocal tract, the propagation of sound can instead be considered through a simple bounded environment like a tube. For frequencies less than $f_{lim} < c/1.71d$ where c is the speed of sound and d is the tube diameter, the vocal tract can be said to have planar wave propagation, meaning that the wavefront propagates with the same velocity at all points on a given cross-section perpendicular to the axis along the tube [20]. Taking an upper bound on the diameter of the tract, the propagation within the vocal tract can be modelled with Equation 2.4 for frequencies below 4 kHz [21]. The simplification of the vocal tract to be a tube of constant diameter is far too simplistic to produce a realistic

output. A better approach is to model the vocal tract as a series of volumes with different characteristic acoustic impedances. As in Equation 2.3, these impedances are dependent on only the density of the air within the tract, the speed of sound in the air, and the cross-sectional area of the volume segment. For a tube which is open at one end, air density and speed of sound should be approximately constant and so the tract can be simplified into a series of connected tubes with different cross-sectional areas.

To finalise this model, only the boundary conditions between volume segments and at the walls of each volume segment need to be accounted for. If these volume segments are considered to be fully bounded in that there is no 'cross-over' area where the impedance is a combination of two regions, then there will be a discontinuity in the characteristic acoustic impedance at the boundary between them. Conservation laws force that the pressure at each side of the boundary is continuous. For Equation 2.3 to hold true, there must be a negative contribution to the velocity term at the interface between two regions. A diagram of this pressure wave scattering at the impedance boundary can be found in Figure 2.1.

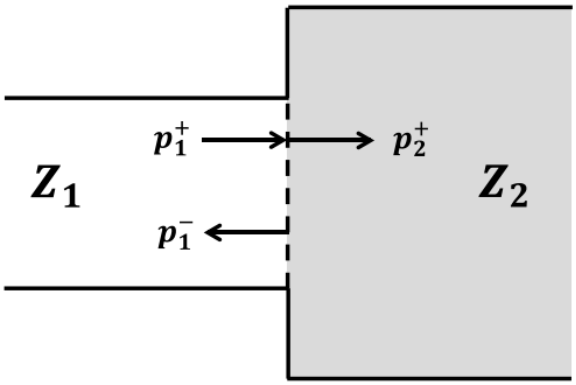


Figure 2.1: Diagram of scattering at an impedance discontinuity. An incident pressure wave p_1^+ on the boundary between Z_1 and Z_2 scatters to create a pressure wave p_2^+ moving through Z_2 and a reflected pressure wave p_1^- moving back through Z_1 [3].

The positive and negative moving components of the pressure wave in the first region, p_1^+ and p_1^- respectively, must sum to the magnitude of the pressure wave in the second region, p_2^+. This creates the reflection relation

$$\begin{aligned} p_2^+ &= (1 - R_{1,2})p_1^+ \\ \text{where } p_1^- &= R_{1,2}p_1^+. \end{aligned} \tag{2.6}$$

$R_{1,2}$ is the reflection coefficient at the boundary between volume elements 1 and 2 and is based only on the characteristic acoustic impedances of the two volumes as

$$R_{1,2} = \frac{Z_2 - Z_1}{Z_1 + Z_2}. \tag{2.7}$$

The resultant pressure through an arbitrary number of volume elements due to an initial pressure wave is as follows:

$$p_n^+ = p_1^+ \prod_{i=1}^{n-1} (1 - R_{i,i+1}) \,. \tag{2.8}$$

Finally, the condition of the end of the vocal tract must be taken into account. The opening at the end of the vocal tract (the lips) may be open or closed. Note that here the other side of the tube (the glottis) is taken to be closed in terms of flow of the acoustic medium. When the lips are closed, assuming the tract walls are fully reflective, the pressure wave is fully reflected back. When open, most of the pressure wave is reflected back but a portion of the pressure wave is allowed to propagate out of the mouth. These conditions affect the types of standing waves that may be formed in the tract. The allowed standing waves have wavelength λ_n and resonant frequency f_n

$$\begin{aligned} \lambda_n &= \frac{4L}{2n-1} \\ f_n &= \frac{(2n-1)c}{4L}, \end{aligned} \tag{2.9}$$

where L is the total tract length, c is the speed of sound, and n is the order of the standing wave [22]. These resonances act to filter the input signal, leading to local maxima in the power spectral envelope of the output signal. The peaks in the power spectral envelope, both caused by resonances in the tract and by other factors such as the properties of the acoustic environment which sounds the tract,

are the formants of human speech [23].

Combining all of the above considerations, the vocal tract can be thought of as a device that takes an input signal u_{in} produced by a volume velocity and transforms it into an output pressure wave p_{out}. This can be described by a Vocal Tract Transfer Function,

$$H(\omega) = \frac{P_{out}(\omega)}{U_{in}(\omega)} \tag{2.10}$$

with P and U as the Fourier transforms of the time domain p and u respectively [24]. This transfer function view of the effect of the vocal tract on the input source is only valid if the vocal tract is a Linear Time-Invariant (LTI) system [25]. The system is linear if the output can be mapped directly to the input meaning that it follows the scaling property and the additive property. The scaling property states that a scaling factor applied to the input is preserved at the output, explicitly $A \cdot U_{in}(t) \rightarrow A \cdot P_{out}(t)$. The additive property requires that the summation of two input signals would result in the summation of their two respective output signals $U_1(t) + U_2(t) \rightarrow P_1(t) + P_2(t)$. The system is time-invariant if the output of an input signal shifted by an amount of time is the same as the original output shifted by that same amount, or if $U_{in}(t)$ goes to $U_{in}(t + \Delta T)$ then $P_{out}(t)$ goes to $P_{out}(t + \Delta T)$. It is not immediately obvious from inspection whether the vocal tract is indeed an LTI system, but insight can be gained by considering a simple analogy for the process of speech production.

2.2 Speech Synthesis

The production of the human voice can be thought of as a filter acting upon a source [26]. The glottis produces two types of source sound relevant to speech: phonation and noise. Phonation is produced by the oscillation of the vocal folds and their repeated collision, forming a periodic wideband waveform known as the glottal flow waveform. This waveform is dependent on the area of the opening of the glottis.

Phonation can be produced artificially by either measuring the opening of the glottis during phonation and playing that signal back, or using one of a variety of source waveform models including the Liljencrants-Fant (LF) model and the Rosenberg model [27, 28].

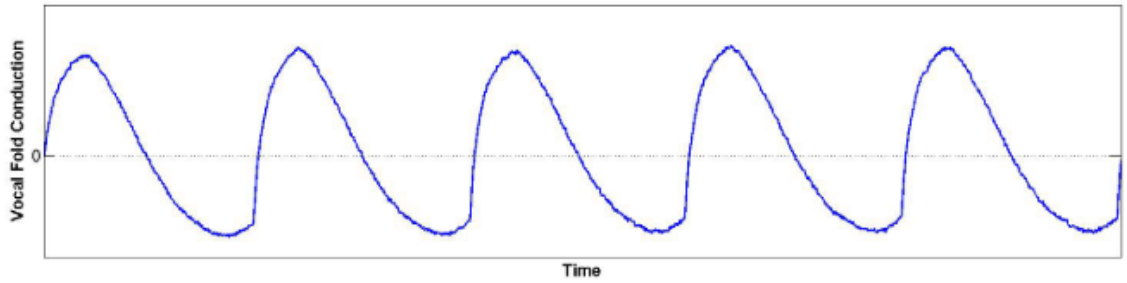


Figure 2.2: Closing of the glottis as a function of time, produced by measuring the conductance of the vocal folds. The average electrical conduction between the probes has been shifted to zero. Higher conduction indicates that the folds are closed [4].

The rest of the vocal tract itself acts as the filter for this system of voice production. The tract filters the input source based on the variation of resonances that form in the tract based on its articulation. In normal speech, the vocal tract is a time-variant system, which moves between different articulations to produce different speech sounds. These articulations cause parts of the tract to change shape with the consequent acoustic filtering being due to the tract shape and size. The coupling of side passages like the nasal cavity also varies, producing anti-resonances when open. Sometimes sounds are produced further up the tract, which would require the resultant source signal to be varied based on the sound being produced. The walls of the vocal tract are also not fully acoustically reflective, impacting formant frequencies and bandwidths. These factors may also suggest that the tract is a non-linear system.

While the source-filter theory in its simplest form would imply that the vocal tract is an LTI system, this theory is an approximation as it is very unlikely that there would be no interactions between the source and the filter itself resulting in a non-linearity [26]. Despite this, the source-filter model is widely used to good effect for reproducing speech and reproduction of the VTTF is a commonly used

comparison point across speech science [29]. As such, despite its inaccuracy to this system, the VTTF will also form the basis for validation for this work, at least during initial studies.

Most research into the creation of systems that can accurately reproduce human speech sounds has a goal of integration within some kind of text-to-speech system. There is a large variety of different text-to-speech implementation methods, varying from applying a filter to a broad input signal based on the required sound [30], creating speech by concatenating different pre-recorded components [31], to using deep neural networks to morph an input signal into the desired output [32]. While these methods can produce convincing artificial speech, it is generally agreed that systems based on modelling the human vocal system fully offer the greatest potential for a natural sounding speech synthesis [33]. Creating these models either requires analogies to the behaviour of the tract [34], or computational modelling methods based on recreating the geometry of the environment to be modelled.

2.3 Geometrical Acoustic Simulation

Geometrical acoustics has roots in geometric optics and is closely related to most modern approaches to graphical rendering [35, 36]. Geometrical acoustic simulations treat the sound waves in the domain as beams or rays and then model the interactions of those objects with the surfaces present in the domain as simple specular collision. These methods typically have the advantage of being computationally fast due to advances made in graphical rendering, very general due to their strictly geometry-based definition, and very simple to implement.

There are three main approaches towards geometrical modelling methods: the image source method, the ray tracing method, and the beam tracing method. A variety of reviews on geometrical methods are available and will be summarised and expanded upon here [37, 38, 39].

2.3.1 Image Source

The image source method involves algorithmically finding all the specular reflection paths between the source and the receiver [40]. A ray from the source that reflects on a boundary before reaching the receiver will create a virtual reflected source, or image source, on a line perpendicular to the surface and at the same distance from it as the original source. An example of an image source can be seen in Figure 2.3, with the source S_c created when rays from source S collide with wall c. The new image source creates a perfect reflection path, as the distance along the straight line between the image source and the receiver is the same distance as between the real source and the receiver. As all reflections are treated as specular, the strength of the signal at the receiver that followed this ray can be calculated using the inverse square law. By algorithmically finding all such rays and their resultant image sources, the respective signals from each source can be summed to calculate the total signal at the receiver for all 1st order reflections. Higher order reflections can also be found by creating secondary image sources from primary ones, and so on [41, 42].

For environments with more complicated geometries than cuboid rooms, it must also be ensured that the image sources are both visible and valid [43]. A source S reflected in a surface A would form image source S_A. If the path between the receiver and S_A cannot intersect with surface A then source S_A is not valid. If an otherwise valid image source cannot draw an uninterrupted path between it and the receiver due to the geometry of the environment, then it is not visible. As all of these image sources are generated algorithmically, it is not always trivial to know if a source is valid and visible, requiring additional computations to be made to check the ray paths for collisions. An example of both valid and invalid image sources can be seen in Figure 2.3. This method is also reported to exhibit errors in environments with corners that have an opening angle that is a non-integer fraction of π [43]. These environments produce solutions in which the total field is not equal to the sum of the direct signal from the source and the signals produced by the image sources. This is due to diffraction effects that must also be accounted for.

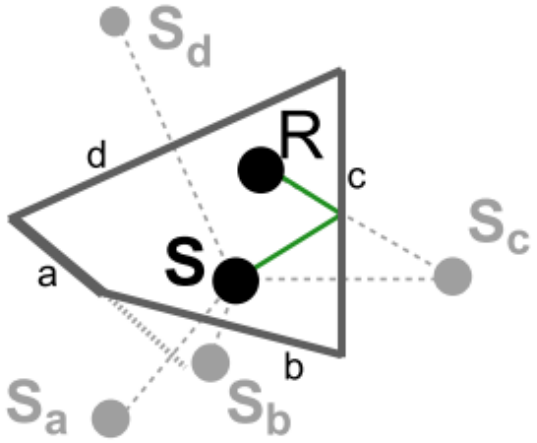


Figure 2.3: An example of the image source method. Here S_b, S_c, and S_d are all valid and visible as they are formed from surfaces that are intersected when tracing a path from the receiver to the source and are not blocked by any other surfaces. S_a is an image source that is not formed by a surface that is intersected when tracing a path to the receiver, so it is not valid [5].

The image source method is simple and very accurate if used in an environment which does not produce any of the errors or complexities previously described, capable of providing an exact solution to the wave equation [44]. However, the issues of visibility, validity, and diffraction that arise in most realistic geometries begin to greatly increase the computational overhead of this technique.

2.3.2 Ray Tracing

The technique of ray tracing is used widely in graphical rendering to accurately model the behaviour of light in a scene. By taking advantage of the similarities between light waves and sound waves, a similar approach can be used for acoustic modelling. In ray tracing, the output energy of a sound source is divided into a set of discrete rays that radiate outwards from the source based on its directionality. These rays propagate at the speed of sound and obey the laws of geometrical acoustics which are essentially the same laws of refraction and reflection as in geometrical optics. Each ray is traced until its energy has dropped below some threshold set at run-time. Interactions with the propagation medium and reflections at medium boundaries are modelled as decays on the power of each ray [45]. By placing a finite-

sized observer within the environment and measuring the properties of incoming rays, a resultant sound can be calculated. An example can be found in Figure 2.4.

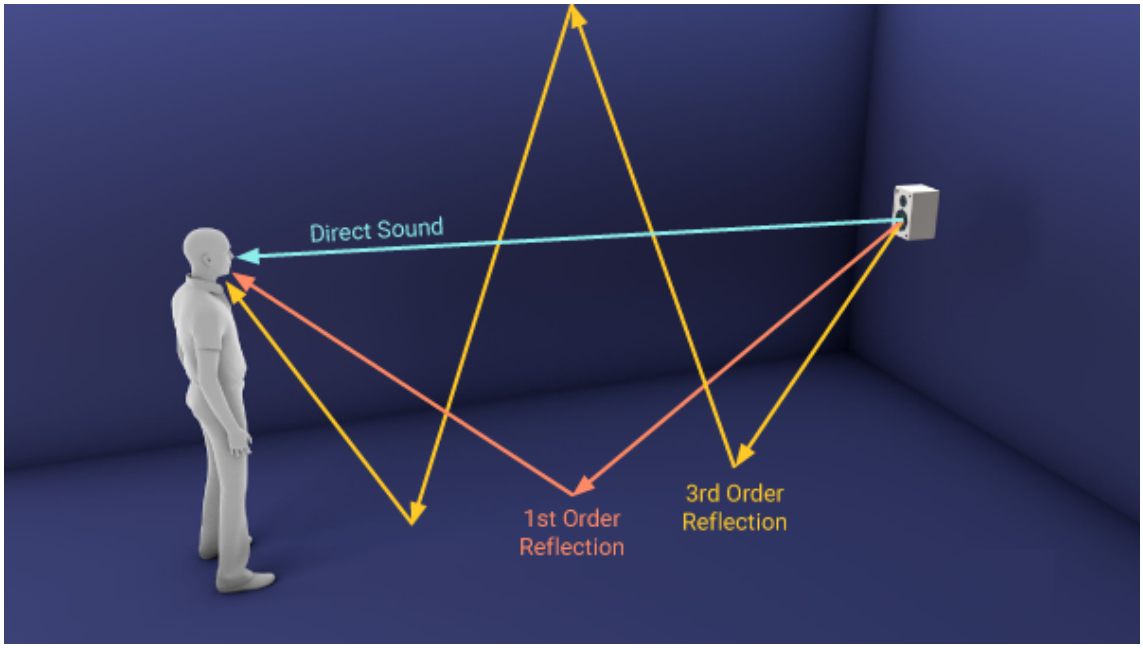


Figure 2.4: An example of ray tracing between a source and a receiver, showing both direct sound and reflections [6].

Ray tracing for sound is fast, efficient, and easy to implement today. It also has an advantage in being completely general with respect to the complexity of the surface geometry of the boundary. However, modelling sound as a ray described only by its power causes a loss of any frequency dependent effects like diffraction, interference, and scattering. Ray tracing is generally also implemented in a stochastic manner, generating a uniform distribution of ray directions across the directivity of the source. If the sample rate is not sufficient then some important ray directions and reflections may be missed from the final solution. In real-time interactive applications, this random distribution is generated on the fly, leading to a varying solution on successive runs.

Ray tracing based methods are still the most commonly implemented ones in virtual reality applications, largely due to a publicly available plug-in developed by Google with support for head orientation, early and late reflections, occlusion, and directivity [6]. Examples of use of ray tracing for sound modelling in literature are available in Kim et al. and Miga-Ziołko [46, 47].

2.3.3 Beam Tracing

Beam tracing is an extension of ray tracing that also originates from the graphical rendering field [48]. In beam tracing methods a set of pyramidal beams are created at the source that occupy all directions emanating from the source, for example a cube made of square-based pyramids with the source at the centre. Each of these beams is extended outwards and objects in the environment are checked for intersection with each beam, starting closest to the source. When an object is found to intersect with a beam, that beam is clipped to remove any volume occluded by the object. Image sources can be formed along with beams representing reflection from an object, and the beam can be cut at the object and extended past it for the case of transmission, as in Figure 2.5 [5, 49, 50]. This set of beams is pre-calculated and then stored for run-time.

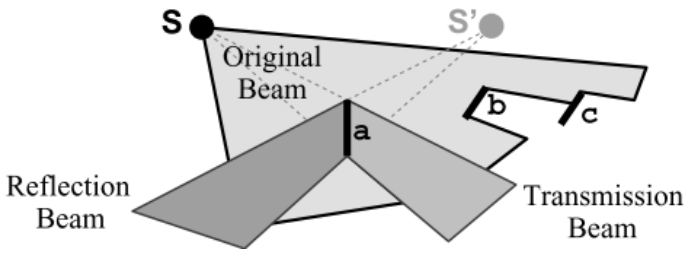


Figure 2.5: Example of a single beam originating at the source becoming occluded by scene objects [5].

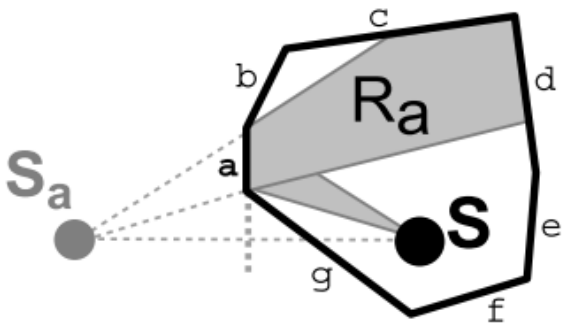


Figure 2.6: Diagram of beam reflection at a surface [5].

As an improvement to the ray tracing method, the number of computations to obtain a resultant signal are drastically reduced as all points within a beam share the same properties. This means the sound at a point can be easily calculated during

run-time by checking which beams the listener is within and summing the contributions of each to the resultant sound. As such, beam tracing also solves the problem of finite sampling inherent to ray tracing methods: As long as all beam paths are found and calculated, any object within a beam is inherently accounted for without needing a potentially arbitrarily large number of ray casts. The pre-calculation does however need to be re-done whenever the geometry of the environment changes or the source moves. This method also shows improvements over the image source model in that it greatly simplifies the amount of image sources that need to be generated for higher orders of reflections. As in Figure 2.6, if a beam incident on surface a creates beam R_a and image source S_a then the only surfaces that need to be considered for subsequent reflections from S_a are those that the beam collides with, namely c and d.

The initial computation of these beam paths can be complicated in complex environments with many reflective and transmissive surfaces, and in environments with curved surfaces. In the years since its inception, many of these issues have been largely rectified to allow for accurate modelling in a reasonable time frame [51, 52].

2.4 Wave-Based Acoustic Simulation

To maintain their computational simplicity, all geometrical methods at least partially neglect the wave-based phenomena of sound including diffraction and interference. When the wavelength of sound is small compared to the geometry of the environment, this is a useful simplification that does not affect the accuracy of the simulation too greatly.

The Schroeder frequency is that below which the acoustic response to an impulse becomes dominated by the low frequency contributions, with clearly distinct separate modes which are governed by wave-based phenomena. The Schroeder frequency is given as

$$F_S \approx 2000\sqrt{(T/V)}, \tag{2.11}$$

where T is a measure of the time a sound impulse takes to decay by 60 dB in the environment and V is the volume of the environment [41]. As the size of the environment decreases the Schroeder frequency increases, meaning that a majority of the frequencies modelled in the resultant acoustic response will be below F_S, and the geometrical approach will break down.

In wave-based methods, the wave equations are solved numerically. This means that no reductions of the physics of the environment are required, and the only inaccuracies come from shortcomings of the method used to solve the equations. Many methods of solving these equations exist, including the Finite-Difference Time-Domain (FDTD) method, the Digital Waveguide Mesh (DWM) method, the Pseudospectral Time-Domain (PSTD) method, the Boundary Element Method (BEM), the Finite Element Method (FEM), and the Finite Volume Method (FVM) [37]. Numerical solutions to complex wave equations generally involve breaking the environment down into smaller cells and then applying a variety of boundary conditions to each of those cells to simplify the solution. Wave-based methods are much more computationally demanding that geometric methods, limiting their usage in real-time applications.

2.4.1 Finite-Difference Time-Domain

Finite-Difference Time-Domain (FDTD) simulation steps through a staggered Cartesian grid to calculate the acoustic pressure and particle velocity at every grid position [53]. The grid is staggered as the particle velocity components of the field are offset from the pressure components by a half integer step. This keeps the difference operators that replace the differentials in the propagation equations centred on the points on the grid which maintains second order accuracy of the finite difference approximation. The method begins with a determination of the acoustic pressure at a point $(i\delta x, j\delta y, k\delta z)$ at time $t = l\delta t$ where i, j, k represent the spatial co-ordinates, $\delta x, \delta y, \delta z$ are the size of the spatial discretization steps, l is the time co-ordinate, and δt is the size of the time discretization. Similarly, the particle velocity must

also be determined at the following positions:

$$\begin{aligned}
&\mathbf{v_x}\left(\left(i \pm \frac{1}{2}\right)\delta x, j\delta y, k\delta z\right), \\
&\mathbf{v_y}\left(i\delta x, \left(j \pm \frac{1}{2}\right)\delta y, k\delta z\right), \\
&\mathbf{v_z}\left(i\delta x, j\delta y, \left(k \pm \frac{1}{2}\right)\delta z\right).
\end{aligned} \tag{2.12}$$

Acoustic pressure and particle velocities at $l = 0$ allow us to find acoustic pressure at $l = 1$ which allows a calculation of the particle velocities at $l = 1$ and so on, following the equations 2.13 and 2.14 where ρ_0 is the local air density, c is the local speed of sound, and $\delta x, \delta y, \delta z,$ andδt are the respective discretization step sizes [53]. By choosing an appropriate value for the time step, these equations provide a full solution of the pressure field at any given point in the grid.

$$\begin{aligned}
\mathbf{v_x}^{[l+0.5]}\left(i+\frac{1}{2},j,k\right) &= \mathbf{v_x}^{[l-0.5]}\left(i+\frac{1}{2},j,k\right) - \frac{\delta t}{\rho_0 \delta x} \times \left[\mathbf{p}^{[l]}\left(i+1,j,k\right) - \mathbf{p}^{[l]}\left(i,j,k\right)\right], \\
\mathbf{v_y}^{[l+0.5]}\left(i,j+\frac{1}{2},k\right) &= \mathbf{v_y}^{[l-0.5]}\left(i,j+\frac{1}{2},k\right) - \frac{\delta t}{\rho_0 \delta y} \times \left[\mathbf{p}^{[l]}\left(i,j+1,k\right) - \mathbf{p}^{[l]}\left(i,j,k\right)\right], \\
\mathbf{v_z}^{[l+0.5]}\left(i,j,k+\frac{1}{2}\right) &= \mathbf{v_z}^{[l-0.5]}\left(i,j,k+\frac{1}{2}\right) - \frac{\delta t}{\rho_0 \delta z} \times \left[\mathbf{p}^{[l]}\left(i,j,k+1\right) - \mathbf{p}^{[l]}\left(i,j,k\right)\right],
\end{aligned} \tag{2.13}$$

$$\begin{aligned}
\mathbf{p}^{[l+1]}(i,j,k) = \mathbf{p}^{[l]}(i,j,k) &- \frac{\rho_0 c^2 \delta t}{\delta x}\left[\mathbf{v_x}^{[l+0.5]}(i+\frac{1}{2},j,k) - \mathbf{v_x}^{[l+0.5]}(i-\frac{1}{2},j,k)\right] \\
&- \frac{\rho_0 c^2 \delta t}{\delta y}\left[\mathbf{v_y}^{[l+0.5]}(i,j+\frac{1}{2},k) - \mathbf{v_y}^{[l+0.5]}(i,j-\frac{1}{2},k)\right] \\
&- \frac{\rho_0 c^2 \delta t}{\delta z}\left[\mathbf{v_z}^{[l+0.5]}(i,j,k+\frac{1}{2}) - \mathbf{v_z}^{[l+0.5]}(i,j,k-\frac{1}{2})\right],
\end{aligned} \tag{2.14}$$

As a solution produced in the time-domain, careful consideration of how to handle the frequency-domain boundary conditions that are common in acoustics is required to prevent a need to transform between domains for every time and space step. The particular formulation covered here is also not sufficient for any geometries not well described in a Cartesian system, needing additional formulation and computation time.

Despite these weaknesses, FDTD methods have gained popularity due to being

very simple to implement and parallelise. With a shift in modern work towards performing complex calculations on the Graphics Processing Unit (GPU), one of the primary concerns of methods like this move from the computational time to dealing with inaccuracies that arise from the method itself. Numerical dispersion effects, phenomena present in discretised numerical simulations of the solution tending further and further away from the accurate value over time, are common in FDTD methods. These effects limit the bandwidth that an FDTD simulation is accurate for, limiting its usefulness. By greatly increasing the resolution of the grid on which FDTD is applied, these effects can be almost eliminated, however this also greatly increases the computational time. As such there is ongoing research into improving the robustness against dispersion effects while keeping computational time low. Van Mourik-Murphy [54] explores the use of a variety of schemes providing orders of accuracy (the difference between the exact solution and the simulated one goes as t^n where n is the order of accuracy) up to 16th order accurate and compares them to a state-of-the-art Interpolated Wideband scheme, showing a fourth order scheme which loses 40 % accuracy but runs 8 times faster than that scheme. Hamilton-Bilbao [55] shows sixth order accurate schemes with much higher order rates of convergence in their dispersion relations and lower computational costs.

2.4.2 Digital Waveguide Mesh

The Digital Waveguide Mesh (DWM) method is based directly on physical modelling techniques, which make use of electronic transmission lines with a carefully chosen circuit layout to model acoustic propagation, dissipation, and reflection [34]. The DWM method takes the concept of scattering junctions connected through transmission lines and extends it into multiple dimensions by instead connecting the scattering junctions with multiple unit-length waveguides, four in a 2D Cartesian grid or six in a 3D Cartesian grid, often referred to as standard rectilinear. Instead of modelling the behaviour of the scattering junctions through electrical components, the DWM method uses a multi-stage update equation very similar to that used in

FDTD [56]. This begins with a calculation of the junction pressure

$$p_J(n) = \frac{2\sum_{i=1}^{N} Y_{J,J_{nei}} p^+_{J,J_{nei}}(n)}{\sum_{i=1}^{N} Y_{J,J_{nei}}}, \tag{2.15}$$

where $p_J(n)$ is the pressure in the junction at time step n, $Y_{J,J_{nei}}$ is the acoustic admittance of the waveguide between junction J and its neighbouring junction J_{nei}, $p^+_{J,J_{nei}}(n)$ is the incident pressure along the waveguide connecting junction J to junction J_{nei}, and N is the number of neighbouring junctions. The junction pressure is then used to calculate the outgoing pressure to each adjacent junction $p^-_{J,J_{nei}}(n)$ using

$$p^-_{J,J_{nei}}(n) = p_J(n) - p^+_{J,Jnei}(n). \tag{2.16}$$

Finally, all incoming pressures for the next time step are set equal to the outgoing pressures of the current time step with

$$p^+_{J,J_{nei}}(n+1) = p^-_{J_{nei},J}(n), \tag{2.17}$$

where $p^-_{J_{nei},J}(n)$ is the outgoing pressure from the neighbouring junction to the current junction.

The DWM method has been shown to be completely equivalent to FDTD methods in terms of solution space and so the only differences between the two arise in their implementation, specifically in their treatment of boundaries within the simulation media [57].

2.4.3 Pseudospectral Time-Domain

Pseudospectral Time-Domain (PSTD) aims to be a faster and simpler version of FDTD methods without sacrificing too much accuracy. This method starts with finding numerical solutions to the linearised Euler equations

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} &= -\left(\mathbf{u}_0 \cdot \nabla\right)\mathbf{u} - \left(\mathbf{u} \cdot \nabla\right)\mathbf{u}_0 - \frac{1}{\rho_0}\nabla p, \\ \frac{\partial p}{\partial t} &= -\mathbf{u}_0 \cdot \nabla p - \rho_0 c^2 \nabla \cdot \mathbf{u} - s\delta(\mathbf{x}|\mathbf{x}_s),\end{aligned} \tag{2.18}$$

with c the speed of sound, ρ_0 the mean density, $\mathbf{u}_0$ the wind velocity vector, $\mathbf{u}$ the acoustic velocity, p the acoustic pressure, $\delta(\mathbf{x}|\mathbf{x}_s)$ the Dirac delta function at the spatial vector $\mathbf{x} = [x, y, z]$ and the source position vector x_s, and $s(t)$ the source strength. The components of velocity $\mathbf{u}$ are denoted $[u, v, w]^T$ [58]. This solution treats Equation 2.18 as a system of first order equations in t and evaluates the spatial differentials using a numerical method before then solving the first order system using some numerical iterative method, for example Runge-Kutta [59]. It should be noted that some FDTD methods have also made use of the linearised Euler equations [60, 61, 62].

The method used to solve these equations that varies from FDTD is the use of a pseudospectral method that estimates the spatial derivatives by Fourier transforming in and out of a different basis, using the equation

$$\frac{\partial}{\partial x}\begin{bmatrix}\mathbf{u}(x,y,z,t)\\ \mathbf{v}(x,y,z,t)\\ \mathbf{w}(x,y,z,t)\\ \mathbf{p}(x,y,z,t)\end{bmatrix} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} jke^{jk(x-x')} \times \begin{bmatrix}\mathbf{u}(x',y,z,t)\\ \mathbf{v}(x',y,z,t)\\ \mathbf{w}(x',y,z,t)\\ \mathbf{p}(x',y,z,t)\end{bmatrix} dx'dk \tag{2.19}$$

for x and similar for the y and z derivatives. This method allows for a large reduction in required resolution compared to FDTD, from approximately 10 points per wavelength to a few as 2, and allows for the use of fast and efficient algorithms like Fast Fourier Transform (FFT) to do the majority of the computational work. This particular formulation suffers in situations with discontinuities or boundaries due to their poor treatment within Fourier transforms, however this too can be resolved by instead relying on an eigenfunction expansion expressed as a linear combination of Fourier transforms. For example, propagation over a rigid ground surface may be

solved with

$$\frac{\partial}{\partial z}\begin{bmatrix}\mathbf{u}(x,y,z,t)\\ \mathbf{v}(x,y,z,t)\\ \mathbf{w}(x,y,z,t)\\ \mathbf{p}(x,y,z,t)\end{bmatrix} = \frac{2}{\pi}\int_0^\infty\int_0^\infty \begin{bmatrix} k\cos(kz)\cos(kz')\mathbf{u}(x,y,z',t)\\ k\cos(kz)\cos(kz')\mathbf{v}(x,y,z',t)\\ -k\sin(kz)\sin(kz')\mathbf{w}(x,y,z',t)\\ k\cos(kz)\cos(kz')\mathbf{p}(x,y,z',t)\end{bmatrix} dz'dk \tag{2.20}$$

for derivatives with respect to z. Once these spatial solutions are found, an iterative method is then used to split the time domain into discrete time steps and then iterate the spatial solutions through that time step using equations derived from Equation 2.18. An example of this is the set of equations

$$\begin{aligned}
&\mathbf{q}(\mathbf{x},t_0) &&= \mathbf{q}(\mathbf{x},t),\\
&\mathbf{q}(\mathbf{x},t_i) &&\approx \mathbf{q}(\mathbf{x},t_0) - \gamma_i\Delta t(\mathbf{W}\mathbf{q}(\mathbf{x},t_{i-1}) + \mathbf{s}(t_{i-1})\delta(\mathbf{x}|\mathbf{x}_s)),\\
&\text{for } i = 1,..,6,\\
&\mathbf{q}(\mathbf{x},t+\Delta t) &&\approx \mathbf{q}(\mathbf{x},t_6),
\end{aligned} \tag{2.21}$$

where $\mathbf{q}(\mathbf{x},t)$ is the vector consisting of the spatial solutions of the acoustic volume and velocity, $\mathbf{s}$ is the source vector $[0,0,0,s]^T$, i is the iteration stage, and $\mathbf{W} = \left[-j\mathbf{R}_j^{-1}\mathbf{L} + \mathbf{V}\right]$ with operators $\mathbf{R}$, $\mathbf{L}$, and $\mathbf{V}$ such that Equation 2.18 can be written as $\frac{\partial \mathbf{q}}{\partial t} = \left[-j\mathbf{R}_j^{-1}\mathbf{L} + \mathbf{V}\right]\mathbf{q} - \mathbf{s}(t)\delta(\mathbf{x}|\mathbf{x}_s)$. Computational complexity calculations discussed in Hornikx et al. [58] state that their particular PSTD method was between 4^D and 8^D times as time efficient as a similar FDTD method, where D is the dimensionality of the problem. This method does however still have all the weaknesses of being a time-domain method in translation of frequency dependant boundary conditions, and also struggles with the treatment of multiple propagation media and also of non-reflective boundaries at high frequencies. More information can be found in Hornikx et al. [63], including details of an implementation.

2.4.4 Boundary Element Method

Instead of splitting the volume of the space into Cartesian unit cells and solving wave equations for each cell, the Boundary Element Method (BEM) instead approximates the surface of the volume, including objects within it, as a series of flat surface segments and solves some analogue of the wave equation, which is taken as constant on each of the surface segments, across them [64]. An illustration of the partitioning of the surface can be found in Figure 2.7.

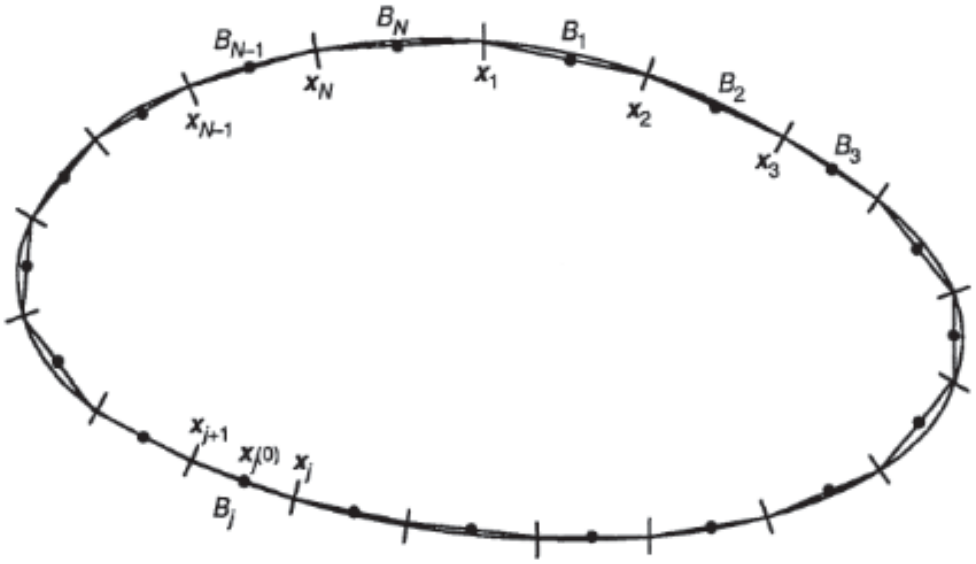


Figure 2.7: A surface split into flat surface segments for the purpose of performing boundary element modelling.

The general method is to replace the partial differential equations that govern the solution in the domain with one that concerns the boundary alone. For example, the Laplace equation $\nabla^2\varphi(\mathbf{p}) = 0$ can be replaced with

$$\int_S \frac{\partial G(\mathbf{p},\mathbf{q})}{\partial n_q}\varphi(\mathbf{q})dS_q + \frac{1}{2}\varphi(\mathbf{q}) = \int_S G(\mathbf{p},\mathbf{q})\frac{\partial \varphi}{\partial n_q}dS_q, \tag{2.22}$$

where φ is the velocity potential, $\mathbf{q}$ is a point on the boundary, $G(\mathbf{p},\mathbf{q})$ is the Green's function for the Laplace equation representing the effect observed at point $\mathbf{p}$ of a unit source at point $\mathbf{q}$, and $\frac{\delta}{\delta n_q}$ is the partial derivative with respect to the unit outward normal at point $\mathbf{q}$ [65]. This formulation completely removes all references to φ at points within the volume and solely relates φ to its derivative along the

boundary.

This set of equations can be greatly simplified by treating the integral equations as integral operators on the boundary functions. In general, if a function ζ is defined on the surface S then applying the following integral on ζ for all points p on S yields a function $\nu(\mathbf{p})$:

$$\int_S G(\mathbf{p},\mathbf{q})\zeta(\mathbf{q})dS_q = \nu(\mathbf{p}). \tag{2.23}$$

Rewriting this integral as an operator gives

$$\{\mathbf{L}\zeta\}_S(\mathbf{p}) = \nu(\mathbf{p}), \tag{2.24}$$

where $\mathbf{L}$ is the integral operator and S is the surface domain of the integration. Writing Equation 2.22 in this operator notation gives

$$\{(\mathbf{M}+\frac{1}{2}\mathbf{I})\varphi\}_S(\mathbf{p}) = \{\mathbf{L}\nu\}(\mathbf{p}), \tag{2.25}$$

where $\mathbf{M}$ is the integral operator corresponding to the left side and $\mathbf{I}$ is the identity operator. By viewing these integral operators as matrices and the boundary functions as vectors, this operator equation can be written as the linear equation

$$(\mathbf{M}+\frac{1}{2}\mathbf{I})\boldsymbol{\varphi} = \mathbf{L}\boldsymbol{\nu}, \tag{2.26}$$

where the components of the vectors $\boldsymbol{\varphi}$ and $\boldsymbol{\nu}$ approximate to the values of $\varphi(\mathbf{p})$ and $\frac{\delta\varphi(\mathbf{p})}{\delta n_q}$ at a set of points on the boundary. As such, if given some combination of these values as initial conditions, a system of linear equations can be constructed and solved to obtain these values across all the surface elements. Finally, once the boundary values are all known, the velocity potential at any point $\mathbf{p}$ within the volume can be computed using

$$\varphi(\mathbf{p}) = \{\mathbf{L}\boldsymbol{\varphi}\}_S - \{\mathbf{M}\boldsymbol{\nu}\}_S \tag{2.27}$$

with all the same notation described above, and then used to calculate the particle velocity and pressure.

With properly defined surface segments and some initial boundary values, the system of linear equations can be solved fairly efficiently using the appropriate algorithms. However, as the complexity of the environment grows, so too does the computational complexity. The number of linear equations also depends on the frequency being modelled, with high frequencies requiring far more equations [43]. While the method described here is general and usually applied to the frequency domain, it can be reformulated into a time domain, iterative process which is theoretically faster but can suffer from numerical instabilities [66].

2.4.5 Finite Element Method

BEM can be thought of as a more specific version of the Finite Element Method (FEM). Both are numerical methods of solving partial differential equations over some volume. Whereas BEM reformulates the partial differential equations such that they only concern the boundary of the volume, FEM instead deals with solving the equations across the volume itself. FEM splits the volume into finite cells defined by a series of nodes, as in Figure 2.8, at which the wave equations are constrained to a finite number of degrees of freedom. Interactions between these volumes is only permitted at these nodes and continuity is required. The requirement for continuity enforces an equality of the free parameters at each node, forming a system of linear equations that can be solved for each node. The variation of the parameters within each finite cell may then be obtained by considering the value at each of the nodes defining the volume [18, 67, 68].

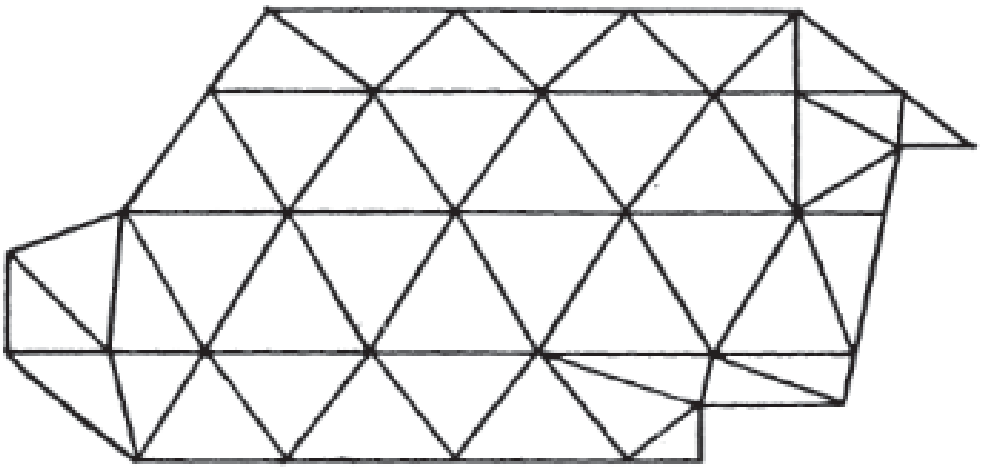


Figure 2.8: Top down view of a 2D cavity divided into cells defined by three nodes at the corners of each cell.

This method is more general than BEM in that it does not require a reformulation of the partial differential equations governing the behaviour of the system, which is not always possible. FEM also easily allows for non-homogenous conditions within the volume by varying the properties within the equations at each node. On the other hand, FEM generally requires a much larger system of linear equations due to the number of mesh nodes generally being significantly larger than the number of surface segments in a BEM simulation of the same volume.

2.4.6 Finite Volume Method

While FEM allows for the defining of cells such that they best suit the environment that is to be modelled this can lead to awkwardness in applying the wave equations at the nodes of each cell. This is common when many cells meet at a node or if a node lies at the boundary of the environment itself. The Finite Volume Method (FVM) is an attempt to combine the geometric flexibility of the finite element method with the flexibility of defining conditions of discreet variables and parameters at various points throughout the environment found in finite-difference based methods [69, 7].

As in FEM, the environment is split into cells defined by a set of nodes. These cells cover the entire volume, leave no gaps, and represent singular definitions of

the space with no overlaps. These cells can be designated such that they form a convenient interpolation structure wherein each node can be interrogated with ease, granting the ability to algorithmically step through the cells by iterating through co-ordinate values. The departure from FEM then comes in the choice of volumes on which the conservation laws are applied. Importantly, these control volumes need not coincide with the cells at all and may even overlap each other if it is deemed useful. The only requirement of the layout of the volumes is that any kind of flux leaving a volume must enter another. Some examples are available in Figure 2.9.

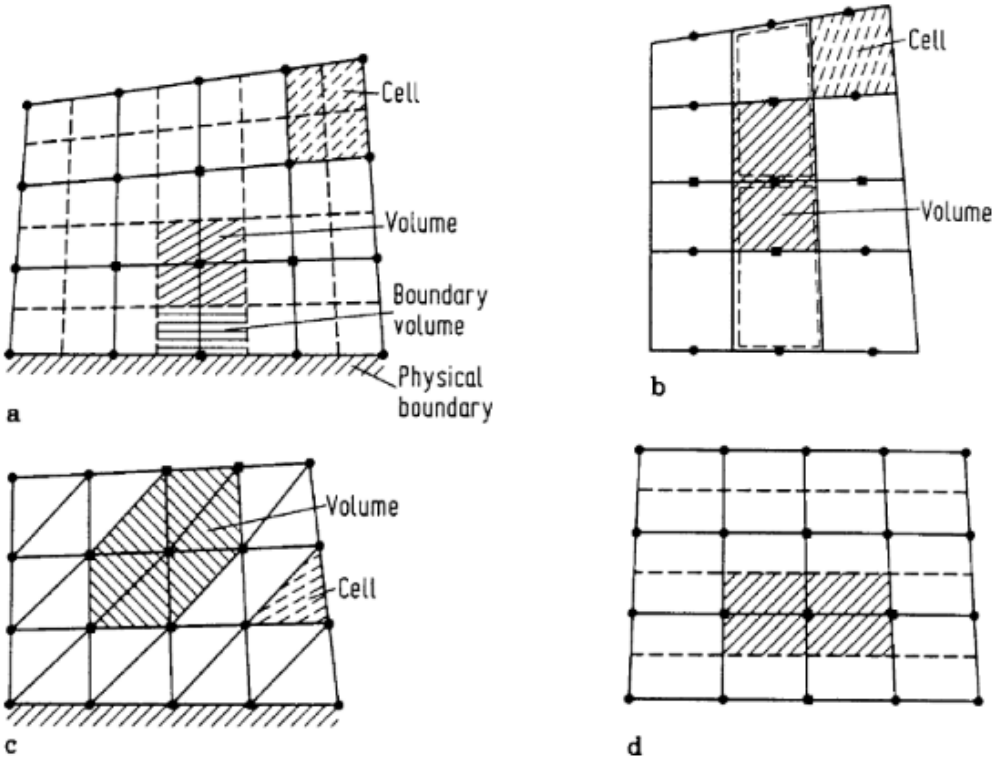


Figure 2.9: Examples of the mismatch of cell definition and volume definition available in the finite volume method. Figures **a** and **d** show volumes that are offset, or staggered, with respect to their cell structure. Figures **b** and **c** show volumes that overlap other volumes [7].

This method, in being a combination of finite-difference methods and FEM, has the advantages of both but also some inherent weaknesses of both. While finite-difference is very powerful when the environment is well described in an orthogonal, equally spaced basis, it becomes very complicated to properly define the partial differential equations that are to be solved when the geometry is less neat. FEM has

the advantage of being able to tightly constrain the degrees of freedom of the wave equations due to the continuity requirements within cells and between them, but as the conservation laws are applied onto the volumes in FVM which are less tightly defined by continuity, it is often not possible to apply these simplifications. The lack of simplification options means that FVM are best suited for simple problems that mainly concern primitive variables, for example those where viscous terms are not dominant. Borrowing from both schemes of geometry definition also enforces a weakness in that FVM struggle with curved surfaces, meaning that FVM cells and volumes are usually constructed from straight lines, limiting the accurate treatment of complex environments.

FVM have also found use as attempts to massage the weaknesses in these two methods where necessary, for example in Bilbao [70].

2.5 Modern Advancements in Acoustic Modelling

Two main approaches to speeding up acoustic modelling are prevalent in recent research: Parallelisation across multiple Graphics Processing Units (GPUs), and hybrid acoustic modelling.

Parallelisation involves splitting up the computational tasks and running them simultaneously on different computing units. The GPU is purpose built for parallelised tasks and so is used to perform parallelised numerical calculations in these methods. As FDTD methods are fully iterative, first iterating over all the grid points to compute initial conditions and then iterating each of those grid points in time, they are very suitable to parallelisation tasks. As such, most of this research uses FDTD simulations. Savioja [71] discusses the implementation of room acoustics models on GPUs, running an FDTD model in real-time for a moderately sized room with a 7 kHz sampling rate, and up to 1.5 kHz for a dispersion error limit of 10 %. Lopez [72] shows the usage of a newer and more powerful Compute Unified Device Architecture (CUDA) to perform FDTD simulations on 134 million node environments over 800 time steps, and 16 million node environments at 44.1 kHz.

Due to the release dates of these papers, they operated on what is now very outdated hardware. As such, modern GPUs would likely to be able to run similar simulations orders of magnitude faster, with the newest consumer GPUs having 4 times the number of CUDA cores and 4 times the memory. This doesn't take into account new advancements in GPU hardware such as tensor cores and new memory architectures.

Even with large performance increases, wave-based simulations on large rooms such as concert venues can be realistically impossible with current restraints on memory and computation time. By limiting the usage of wave-based methods to the low frequencies, where the effects they model are most prominent, and switching to geometrical methods for the high frequency regions, which wave-based models are generally not suited for, it is possible to create a full high performance acoustic model without sacrificing too much accuracy. These hybrid acoustic models have proven effective with a wide variety of numerical modelling methods and geometrical methods, including FEM and FDTD simulation, ray tracing, beam tracing, and image source simulation [73, 74]. While hybrid methods are obviously a powerful combination of the strengths of both methods, they do incur the complexity of both in defining adequate boundary conditions and other important parameters for the numerical models, and in taking into account directivity, material properties, and performing the collision checks required for geometric models.

2.5.1 Acoustics in Virtual Reality

Modern Virtual Reality (VR) makes use of a head-mounted display and often a controller or pair of controllers, all of which are able to move around the real space with those movements mapped into the virtual one. A large proportion of early and even current work in VR is focussed around improving visual fidelity in the way of higher resolution screens, wider field of view, higher refresh rates, and so on. However, even in work that predates the advent of modern VR systems, the importance of a realistic recreation of the audio environment was understood, sometimes even said

to be more so than the visual stimuli [75, 76, 77]. While accurate sound is invaluable to a truly immersive virtual environment, it is a task which adds a large amount of processing time to an already complicated field.

As the user's vision is fully immersed within the virtual environment, anything that leads to behaviour of the visual system in a way that isn't realistic can immediately lead to a break-down of immersion, or even the onset of nausea. The main example of this is long computation steps during run-time leading to the inability to run the virtual environment at a consistent frame rate, causing stutters in both visual and audio production. Not only do VR applications have to perform some kind of accurate model of sound within the environment, but they must also do this in real-time while allowing for the position of the head and the direction it is facing to change during the simulation. The solutions to this challenge fall into two categories: pre-calculated and real-time approaches.

Pre-Calculated Audio

In these systems, the vast majority of the computation of the acoustic environment is done in a pre-calculation phase, with the pre-calculation only needing to be adapted during run-time to account for the users position and orientation within the environment. As this method only requires relatively simple calculations be done each frame, it lends itself to more accurate wave-based methods.

One sensible way of constructing this system is by calculating the room impulse response in a grid of receiver locations across the entire environment and using it to generate a resultant sound field at each point prior to run-time. As a user moves around the virtual environment, the sound field at the point they are closest to could then be played back to them, spatialised correctly to their orientation. For this method to be viable, the solutions generated during the pre-calculation phase must be stored in a manner that contains spatial information. This can be done by mapping the solution onto a set of spherical harmonic basis functions to provide a functional form for the sound at any given point [78]. To perform the spatialisation,

the calculated response from the model needs to be rotated into the same frame of reference as the users head. This can be done using a Head-Related Transfer Function (HRTF) that modifies incoming sound for the directivity of the head, and for the effects created by the scattering of sound off the body [79]. By using an HRTF that is given in the form of coefficients of the spherical harmonics, this becomes a relatively simple task [80, 81].

This method is intuitive and effective, allowing for wave-based acoustic modelling techniques to be used to produce real time spatialised audio. Unfortunately, as the room impulse responses are calculated before run-time, each unique source location or directivity would need to have a full set of room impulse responses generated, which greatly increases the memory requirement to store them and makes fully dynamic sources impractical

By reformulating the problem, having both dynamic sources and dynamic receivers are possible. The Equivalent Source Method (ESM) considers the fact that the way an object interacts with the acoustic environment is fixed regardless of the exact properties of the sound field. ESM calculates a per-object and inter-object transfer function for each object that describes its interactions with the acoustic environment [82]. The per-object transfer function describes the effects of the object on incoming sound, encapsulating behaviours like reflection, scattering, and diffraction. This function then maps an incoming sound onto an outgoing one. The inter-object transfer function captures the effects of the outgoing sound field from one object on another, mapping the outgoing sound field into the incoming one of another object. For each object, the per-object transfer function is represented by the scattering matrix $\mathbf{T}$, the inter-object transfer function to be represented by interaction matrix $\mathbf{G}$, the sound field emitted by an arbitrary source to be vector $\mathbf{S}$, and then solve

$$(\mathbf{I} - \mathbf{TG})\mathbf{C} = \mathbf{TS}, \tag{2.28}$$

where $\mathbf{I}$ is the identity matrix, as a linear system of equations for $\mathbf{C}$, the strength

vector of the propagated sound field for the entire scene. As long as the objects and their properties remain static, **T** and **G** are constant for each object. By noting the principal of acoustic reciprocity, which states that one can reverse the source and receiver in a scene without changing the acoustic response, it can be seen that only **S** varies for dynamic sources and receivers. By reformulating Equation 2.28 as

$$\mathbf{C} = (\mathbf{I} - \mathbf{TG})^{-1}\mathbf{TS} = \mathbf{DS}, \tag{2.29}$$

where $\mathbf{D} = (\mathbf{I} - \mathbf{TG})^{-1}\mathbf{T}$, Equation 2.28 is simplified to an operator acting on a source/receiver. This operator can be pre-calculated and then applied to **S** during run-time. This method is more computationally demanding, requiring the propagation equation be solved during run-time, but fully allows for dynamic sources and receivers.

Other methods are also under active research, including an image source method that allows for dynamic receiver locations based on pre-calculated image sources [83], which will not be fully discussed here.

Real-Time Audio

Real-time approaches attempt to fully simulate the audio of the environment during run-time. This method doesn't impose any restrictions on source or receiver positions, and doesn't require the positions of either to be fixed or limited. Due to the requirement within real-time schemes to perform the audio simulation with a reasonable update rate, wave-based methods are largely ruled out in favour of geometrical approaches like ray tracing.

One of the most computationally difficult tasks in most geometric approaches is performing intersection tests in real-time to determine which faces a ray or beam collides with. These tests need to be performed whenever the source or receiver moves or changes its properties so that the acoustic environment stays up to date. These tests become more complicated with the number of surfaces in the environment, so a common simplification is to reduce the number of surfaces by combining

similar faces [38]. A further simplification can be made by accounting for the sensitivity of the human ear to the different reflection depths, updating direct sound and low order reflections with a rate of 25 Hz to 100 Hz and updating high order reflections with a rate of 1 Hz to 5 Hz [38, 84]. This is due to direct sound and low order reflections changing more significantly with small changes in positions than high order ones.

The incoming orientation of sound is well known due to the ray structure, so spatialisation of the sound to the user using a HRTF is quite simple, if still computationally taxing. The convolution can be made more efficient by using an HRTF that is described in spherical harmonic basis functions as mentioned in Section 2.5.1. This has been shown to have a minimal negative impact on simulation accuracy with a reduction on compute time [85].

2.6 Chapter Summary

This chapter has presented a theoretical grounding from which to approach the acoustics of speech production throughout this work. A simple approach to vocal tract acoustics, from conservation arguments and the wave equation to the concept of the transfer function and its application as a method of speech synthesis. While there is a great deal of detail and rigour which could be added to this description, it is sufficient to understand speech to the level which is required for this work.

In addition, contained within this chapter is a broad review of acoustic modelling techniques including both techniques based on geometric acoustic principles and techniques which aim to directly solve the acoustic wave equation. The models are presented in full and with sufficient mathematical rigour to perform a basic implementation of their respective processes. The advantages and disadvantages of the methods have been discussed, comparing between the two schema of acoustic models. For maximised accuracy and alignment to the true physical regime within the tract, wave-based methods have been deemed as the ideal way to simulate acoustic propagation in this work.

A brief introduction to some of the modern advancements in these methods is presented at the end of the chapter, including the use of sophisticated and purpose-targeted hardware for parallelisation, and the applications of acoustic simulation in Virtual Reality (VR).

Chapter 3

Pre-Existing Acoustic Modelling Packages

Real world applications will place requirements on the accuracy simulated acoustic propagation data based on their usage. To ensure that accuracy of simulated data is high, the propagation in the tract must be modelled with high physical accuracy with minimal simplifications and compromises, and must contain all of the important features of the acoustic field. As discussed in Chapter 2, both geometrical and wave-based acoustic modelling packages are capable of producing accurate results when applied to problems they are well suited for. Equation 2.11 presented the concept of the Schroeder Frequency: the frequency below which the acoustic response of the space is dominated by wave-based phenomena like resonances and diffraction. The Schroeder frequency is dependent on both the measure of the time taken for a pressure impulse in the space to decay by 60 dB, and on the inverse of the space's volume. Taking only the relationship of the Schroeder frequency and the volume into account, for large spaces F_s will be very small and the vast majority of the frequency range which is relevant to the human ear will be governed by geometric behaviour. For small spaces the inverse is true.

The Schroeder frequency was originally conceived for room acoustics applications though. In small highly reverberant spaces will likely always produce a overestima-

tion of the cut-off point for the transition between the dominance of wave-based phenomena, the modal region, and the dominance of more geometrical behaviour, the statistical region. The tract is more likely to behave based on the Schroeder frequency for a duct, which is given as

$$f_{rect} \approx 0.2721 \frac{EDTc_0^2}{S} and f_{circ} \approx 0.6511 \frac{EDTc_0^2}{S}, \tag{3.1}$$

where EDT is the "early decay time", c_0 is the speed of sound in the medium, and S is the surface area of the end of the duct [86]. The EDT is the time taken for initial quasi-linear acoustic energy decay after an excitation to transition into the slower non-linear late decay. This is a more accurate description of the cut-off for the transition between the modal and statistical regions but is still based on much larger geometries than the vocal tract and does not account for the high levels of damping in the vocal tract.

Measurements performed using 3D modelling tools on models that will be used for simulation in this research place the volume of the vocal tract during certain articulations to be less than 100 cm^3. While the decay time will likely also be very short many sources of damping, the acoustic field inside the tract is likely well within the modal region

Human vocalisations generate acoustic frequencies with relevance to speech perception up to approximately 22 kHz, with the frequencies up to 5 kHz being often stated as those that are most important for speech intelligibility [87]. As such to accurately model this 'low frequency' acoustic propagation, some treatment of the wave-based properties of sound will be required.

In this chapter, a variety of modelling packages have been tested and explored for potential use as the acoustic propagation solver in this work. Their advantages and disadvantages within the scope of this work are also discussed. Vocal tract models used during these studies are from the same corpus produced by Speed et al. [88] and will be discussed further in Chapter 4.

3.1 Resonance Audio

Despite the issues discussed with the Schroeder frequency of the vocal tract, the field of geometric modelling of acoustics in real time 3D applications is much more senior than wave-based modelling in those same applications, so one popular package will be explored for posterity. As discussed in Section 2.3.2, Google's Resonance Audio plug-in is a very popular method of simulating sound in 3D spaces due to its implementation within both of the commonly used 3D game engines Unity and Unreal [6].

Resonance audio uses a geometric-based simulation method, with special care paid to effects that would negatively impact immersion for a user, including source directivity, occlusion, and reflections. The plug-in allows you to pre-calculate the acoustic properties of a space using probes that can be placed around the environment. Before run-time, resonance audio will fire rays from these probes and calculate beam paths and room properties which are then used during run-time to modify and filter sounds the user is hearing.

A vocal tract model was imported into the Unity game engine and a series of resonance audio probes were placed inside the tract at points where the sound field was deemed to vary greatly from the position of the last probe. As the probes store the acoustic properties of the space, the closest probe to the user is the one that will govern the way they hear the sound. Placing probes at acoustically varying locations, for example at constrictions and the opening of side passages, will best capture the variation of sound throughout the tract. After placing the probes, the pre-calculation of acoustic properties was performed before running the program which allowed a user to move through the vocal tract in 3D space and hear how a sound, produced by a source placed at the glottis, changes throughout the tract.

During run-time, an issue with the implementation of resonance audio was encountered. To ensure that the acoustic properties of the space are properly applied, resonance audio probes will only apply their modifications to sound sources within their sphere of influence. As such, acoustic effects were only being applied to the

sound source when in the region of the probe closest to it. To attempt to deal with this limitation, a single large probe was placed halfway down the tract such that the entire tract was within its sphere of influence. In this set up though, the resultant sound was mostly constant when moving around the space implying that this method is not suitable.

In both layouts, the effect on the sound was largely dominated by ballistic reflections of the sound off of the walls of the tract, leading to an unrealistically reverberant sound. This is likely due to incorrect material properties for the walls of the tract, causing them to be too acoustically hard. However, even when an acoustically soft preset material, in this case curtains, was chosen for every surface of the model it was still very reverberant. As a geometric model, instead of the vocal tract being a space with small constrictions and diffraction effects, resonance audio interprets it closer to a large enclosed conference hall. Geometric methods have always been expected to be unsuitable for this task, and this method was investigated purely for posterity and as a contrast to the wave-based methods also discussed in this chapter.

3.2 openPSTD

openPSTD uses the pseudospectral time-domain method described in Section 2.4.3 to compute sound propagation in built environments [63]. While this software suite is designed specifically for architectural acoustics, it is a general PSTD simulation and as such would be a suitable simulation method. openPSTD is available as a plug-in for the 3D modelling software Blender which is capable of running plug-ins written in Python directly inside the software. This allows for complex geometries to be made very easily using the pre-existing tools available in Blender. An example of acoustic propagation produced by openPSTD can be seen in Figure 3.1.

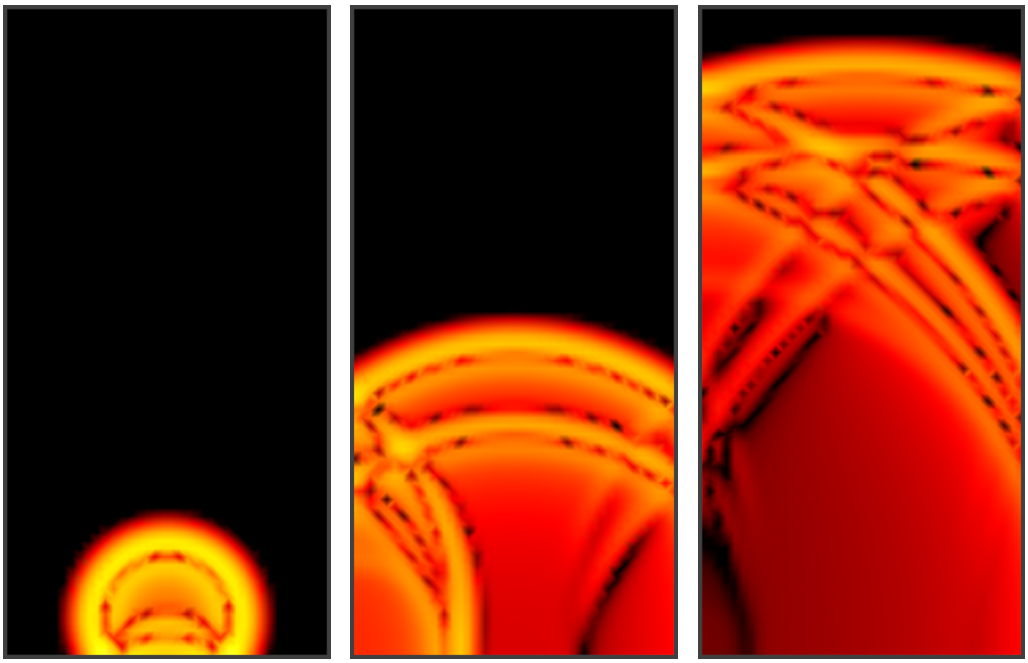
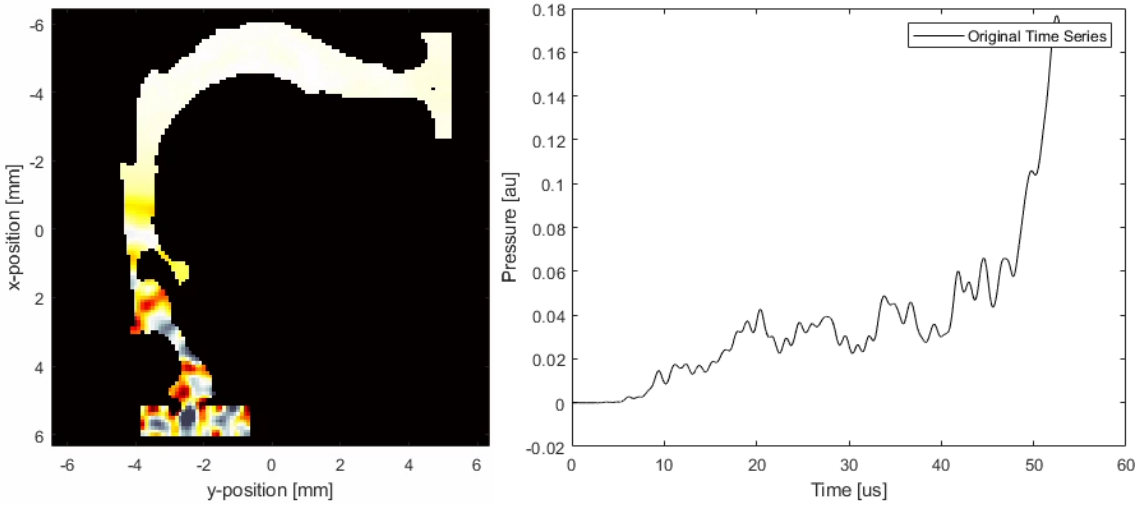


Figure 3.1: Sound propagation through homogenous domain with semi-absorbing boundary conditions at increasing time steps. Simulation results from openPSTD. More luminous colours represent a greater magnitude of acoustic pressure.

openPSTD also produces impulse responses based on receivers placed in the geometry, making it very useful for storing the acoustic propagation for future playback. While openPSTD is an effective modelling package that has been used in many publications and applications, it is currently limited to 2D geometries comprised solely of rectangular elements. 3D simulation is required in this work to corroborate the assertion that its outputs are truly accurate to the real behaviour of sound in the vocal tract. While 3D geometries can be converted into structures comprised of cuboids, the effects of non-physical boundaries on the accuracy of the output of a theoretical 3D version of openPSTD would need careful consideration. Still, the implementation of absorbing boundary layers in this package would make it an appealing method if it was upgraded at some point in the future.

3.3 k-Wave

k-Wave is an acoustic toolbox for MATLAB which makes use of a frequency domain pseudospectral method with accommodations for general geometry and varying media [89]. It supports domains up to 3D however the 2D scheme will be focussed on here as importing geometry in 3D was not trivial. In 2D, geometry can be imported very simply as an image file, where the brightness of pixels determines the acoustic allowance of that region. White pixels are interpreted as free space and black ones are interpreted as walls. After providing the geometry and specifying the source and receiver locations, running the simulation is simple.



(a) Pressure propagation through 2D vocal tract after approx. 20 µs.

(b) Plot of pressure against time in µs at the receiver placed at lips.

Figure 3.2: k-Wave simulation output both as a 2D plot and a pressure against time graph. In the left plot, the colour scale goes white-yellow-red-black with darker colours representing a higher pressure. Note that for simulation speed, the vocal tract was imported at roughly 1/10th scale.

k-Wave is capable of producing animations of pressure flow through the simulation medium, the raw data for the acoustic pressure at the receiver location, impulse responses, and much more. Figure 3.2b shows simulation data produced by k-Wave

with a receiver placed at the black pixel in the top right of the Figure 3.2a. The left figure shows acoustic pressure at an instant shortly after the simulation begins, with darker colours showing higher pressures. The right figure shows the pressure at the listener node increasing over the course of the simulation as the acoustic field propagates. In this simulation, the speed of sound in the walls of the tract and the density of the tract are defined based on those same properties in air scaled by a fixed value. This produced a speed of sound in the medium of $6860\,\mathrm{m\,s^{-2}}$ and a density of $24.5\,\mathrm{kg\,m^{-3}}$. These values were purely for testing and do not represent accuracy of those two variables, instead having been chosen to minimally change the example simulation routine provided by k-Wave. Figure 3.2b shows the effect of these variables in the appearance of heavily attenuated acoustic pressure around 5 µs at the receiver, which is appropriate for transmission through the solid walls, before the much higher amplitude pressure flow through the tract around 50 µs.

While this behaviour is accurate to the provided vocal tract model, creating a version of this simulation which is accurate to the human vocal tract would require accurate modelling of the geometry of the head around the tract and of the varying materials which make up the walls of the vocal tract, the throat, and the head. This process would technically be required for any accurate simulation of pressure flow in the vocal tract. Importing all of this data into k-Wave to perform the situation would be cumbersome in 2D, and become significantly complex in 3D. k-Wave is clearly a very sophisticated acoustic simulation toolbox which given enough effort would be suitable for the simulations required in this work, however as discussed before a core goal of this research is for the acoustic simulation process that is ultimately chosen to be accessible and useable by individuals with limited technical knowledge and resources. k-Wave is a package built on top of MATLAB which does not offer an unlimited free licence at this time. As such k-Wave will not be used in this work.

3.4 COMSOL

COMSOL is a commercial modelling suite that can be used to model a wide variety of physical systems, including coupling between different fields of physics, for example acoustic pressure flow and fluid dynamics. COMSOL provides a variety of modules that include more powerful tools for interrogating different physical systems. The acoustic module includes FEM solvers, BEM solvers, a time-explicit discontinuous Galerkin method solver, and ray-based methods. COMSOL also features full support for 3D modelling within the software, vastly simplifying the process of making a model 'computer readable' discussed in Section 4.1.2. All of these features and properties make COMSOL a very attractive option for performing the acoustic modelling required for this project.

3.4.1 Boundary Element Method in COMSOL

The first method that was explored was the frequency-domain boundary element solver. This method was chosen as it is theoretically very compatible with the input model. As will be discussed thoroughly in Chapter 4, these models are generated from Magnetic Resonance Imaging scans by performing a flood fill on the empty space inside the vocal tract and then drawing boundary faces around the point cloud produced during that fill. This gives us an STL model comprised of N-sided polygonal faces. These faces should be able to be interpreted by the software as the boundary elements of our simulation medium: the air within the tract. COMSOL is also capable of setting complex boundary conditions on a face-by-face basis that can include frequency dependent damping effects.

COMSOL requires that models imported for simulation will be capable of producing an adequate result. This is most noticeable in a minimum face size. In most wave-based modelling schemes, the simulation medium needs to be split into some kind of mesh that the algorithmic solver can operate on to produce a result. If the faces of the imported model are too small, then the subroutine that creates the mesh will not be able to accurately form the mesh around that face without clipping

through parts of the model or other sections of the mesh. The models created from the MRI scans are extremely complex and have many artefacts that have not previously been relevant in their usage. However, these artefacts make it very difficult to import these models into COMSOL. Given enough time, the models could likely be manually prepared for COMSOL usage, but it would require some non-predictable reduction in the accuracy of the geometry of the vocal tract in the removal of very tight constrictions. To ensure reproducibility, a programmatic method of preparing these models would be preferred. One such programmatic method of simplifying 3D geometry is that of voxelisation, to be discussed thoroughly in Section 4.1.2. While one of the draws of using COMSOL was the ability to directly use vocal tract models, it is still a very sophisticated modelling package that may produce more accurate results than a hand-made simulation.

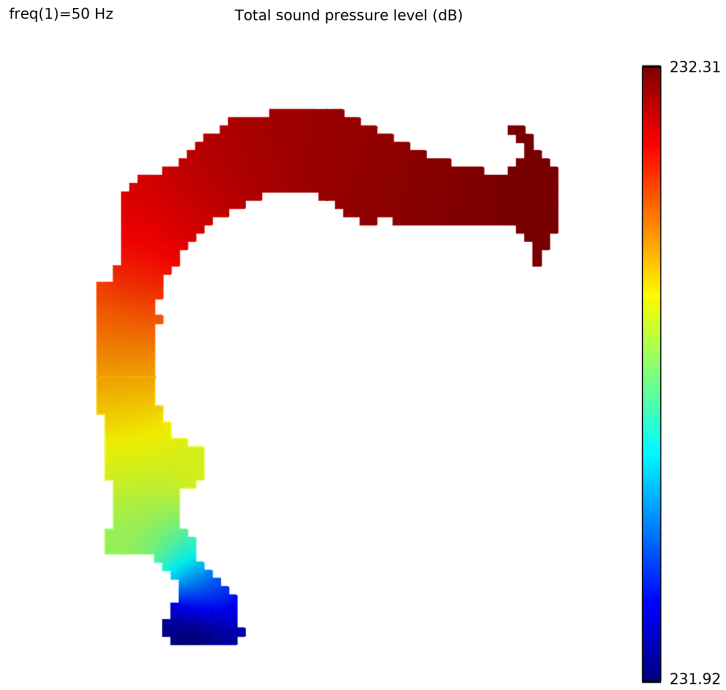


Figure 3.3: Plot of total sound pressure level throughout vocal tract at the default frequency of 50 Hz over a period of 7.5 ms. Data produced using frequency domain BEM solver in COMSOL. Due to unphysical import parameters, sound pressure level is not correctly scaled.

Figure 3.3 shows the sound pressure level throughout the tract. In this particular simulation, there was no absorbing boundary condition at the lips (top right) or at

the glottis (bottom left). The tract therefore acts as a closed tube with acoustic pressure build up at the end opposite the source. Adding an absorbing boundary condition at the lips was possible, but was not explored due to time restrictions.

3.4.2 Time-Explicit Simulations in COMSOL

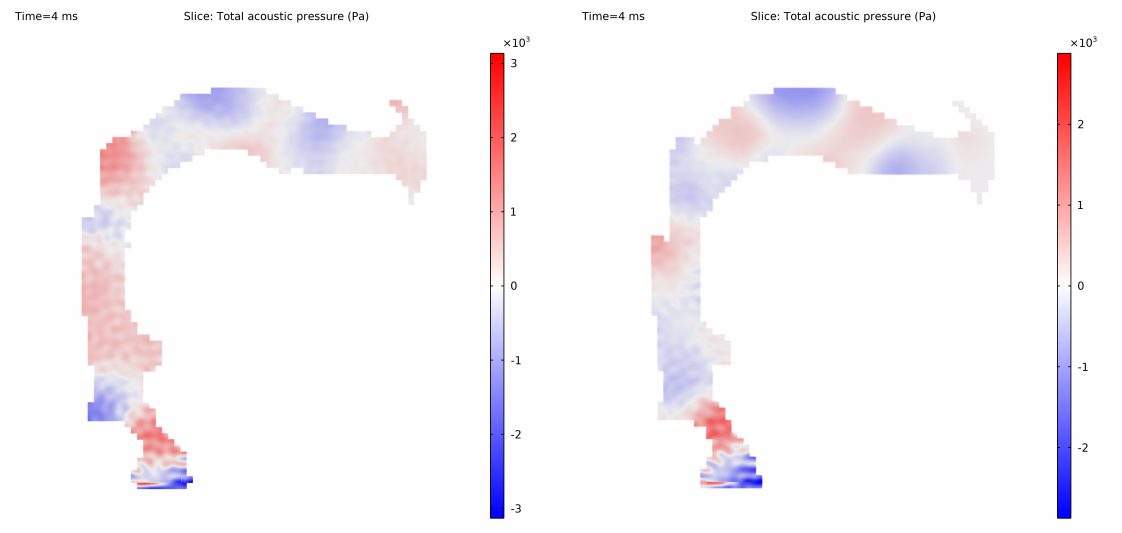
To be more comparable to the simulation methods being explored simultaneously to this work in COMSOL, the Time-Explicit Discontinuous Galerkin (DG) solver was more thoroughly explored. The DG method is another system for solving the differential wave equations. It is a finite element method, operating on a mesh of discrete regions, which uses fully discontinuous basis equations such that each element only communicates with its immediate neighbours regardless of the order of accuracy of the scheme. This provides a great deal of flexibility in the shape and layout of the mesh, the ability to use a very localised data structure, and consequently a very high efficiency in parallelisation. As a time explicit method, the DG solver also produces a value of pressure at every mesh element in the model at every time step which allows for a direct comparison with FDTD style methods in terms of pressure at some pre-defined 'listener' location.

As before, the source is modelled with a vibrating surface at the glottis and the visible region in the plots found in Figure 3.4 is the simulation medium, in this case air. In both the closed and open systems, in addition to the regions of high and low pressure moving through the tract, small-scale vibrations in the pressure can be seen within the tract most clearly close to the glottis. These vibrations are unexpected, and have a few probable causes arising from the properties of the simulation.

We observe that the vibrations die out approximately halfway through the tract when there is an absorbing boundary condition at the lips. This implies that reflections likely play a large part in the source of these vibrations. The walls of the simulation medium are either modelled as acoustically hard with some power loss upon reflection, or as fully absorbing in the case of the lips. The hard surfaces are likely not sufficiently accurate to the walls of the real vocal tract but without further

measurement it remains to be seen if these vibrations are accurate to the real world or meaningful to the output.

Another possible cause of error in this simulation is the use of a voxelised model. In Section 2.4, the dominance of low frequency modes in the acoustic field below the Schroeder frequency was discussed. Diffraction occurs when waves pass through constrictions much smaller than their wavelength, or past impeding obstacles like edges or corners. The voxelised version of the vocal tract model introduces many non-physical edges and corners for waves to diffract off of, which would cause local vibrations in pressure whenever it occurs. As the solver only generates a pressure in a cell of a mesh based on its neighbours, the effect of these edges would be dependent on the exact mesh generated by the software. Again further measurement would be required to determine whether the proposed diffraction off of voxelised surfaces has a noticeable impact in the output sound field.



(a) Acoustic pressure with no absorbing boundary at the lips.

(b) Acoustic pressure with a fully absorbing boundary at the lips.

Figure 3.4: Plots of acoustic pressure produced using the Time-Explicit Discontinuous Galerkin (DG) solver in COMSOL. Both plots shown 4 ms into simulation.

3.4.3 Limitations of COMSOL

COMSOL is an industry standard in physics simulations for engineering applications. As a commercial tool, COMSOL does not offer a free access licence. All the work in COMSOL as part of this research was done during a two-week trial period. A simulation output time of 7.5 ms was chosen in this study as a starting point due to the standard 10 ms length of an LF model pulse, to provide a reasonable view of the response of the tract over a similar time period. Producing this data required a simulation run time of approximately 60 h. This time may have been possible to reduce given more familiarity with the tool, however that was not available within this time frame.

While COMSOL does offer the ability to run simulations on dedicated cloud servers with specific licences, this simulation length still completely rules out COMSOL for use as part of interactive applications which are a core aim of this research. A much shorter simulation time can still be used to produce a transfer function that represents the frequency response of the tract, however care needs to be taken to ensure that the simulation time is long enough to see all the effects of the geometry of the tract. Assuming linear time scaling, a 1 h COMSOL simulation of this tract would still only produce 0.125 ms of pressure data which would not be sufficient to model the behaviour of the tract. The simulation times for the DG solver are as long, if not longer, than those for the frequency-domain methods included in COMSOL.

While COMSOL is an incredibly sophisticated physics simulation package, which could produce extremely accurate outputs by taking into account other physical regimes such as fluid dynamics into the simulation, its price and run time are not suitable for the scope and aims of this project.

3.5 Chapter Summary

The simplest way to perform an accurate acoustic simulation in this work would be to make use of one of the many available software packages that are designed

with this purpose in mind. A small subset of the most promising and applicable available packages for acoustic simulation are presented and explored throughout this chapter.

Google's Resonance Audio plug-in provides a simple-to-implement avenue for creating realistic sound outputs in arbitrary acoustic environments, but as a geometric method it breaks down quickly in tight confined spaces such as the vocal tract. k-Wave and openPSTD are strong tools written in prominent scientific programming languages with their own shortcomings that limit their use in this work. k-Wave supports arbitrary geometries and acoustic transmission in the walls, but geometry import especially in 3D is highly complex and the package is only available in the licence based MATLAB language. openPSTD solves the wave equation in the time domain and within 3D modelling software, but currently only supports 2D simulations.

Finally, COMSOL was explored as a feature rich multi-physics modelling suite. While COMSOL is certainly capable of producing exceptionally accurate acoustic data with a variety of different simulation methods and measurement schemes, its complexity is much higher than what is needed for this work which leads to simulation times far longer than are conveniently useable here. As a commercial paid package, it is also highly inaccessible to non-commercial individuals that would need to pay for licences.

The lack of a suitable modelling package enforces the need to produce a bespoke acoustic modelling package that is designed directly for the needs of this work.

Chapter 4

Acoustic Propagation Algorithms in Python

Chapter 3 described a number of popular and powerful modelling packages, however none of the packages which were explored were deemed as satisfying all the specifications for this research. The two most important requirements for the acoustic simulation used in this work are: the simulation must be reasonably accessible in terms of not requiring much technical knowledge or software that is difficult to run or expensive, and the simulation must be able to produce output continuously in space and time throughout the tract to a level of accuracy that is at least comparable with prior research. Guaranteeing that both of these requirements are met is most simply accomplished by creating a bespoke acoustic propagation algorithm which is designed to be easily useable and is written in an open source language.

Creating this simulation algorithm will require careful consideration and research into how some existing tools solve some difficulties with this simulation process, like translating 3D geometry into something computer readable, producing output from the simulation in a way that can be used for a variety of applications, and how to perform analysis on that data for comparison with previous research and general validation of accuracy. As a goal of this work is to present this data in an intuitive and understandable way, the requirements of the simulation in terms of visualisation

may help drive some of the decisions made in the design of the simulation itself.

4.1 3D Visualisation and Modelling

Initial work on this project investigated both simulation techniques and visualisation methods simultaneously. Two major challenges were quickly identified that were greatly influenced by the proceeding research into visualisation methods: ensuring that the human vocal tract is recreated within the simulation accurately enough to produce accurate sound, and being able to present that sound throughout the vocal tract in an accessible and intuitive way.

The first of these two challenges will require proper pre-processing treatment of the available vocal tract models, and some effective method of digitising them for an algorithmic simulation. The second challenge will require thought and experimentation in the different ways of presenting the data, likely concluding in presenting them in a 3D virtual space.

4.1.1 Pre-Processing of 3D Vocal Tract Models

The vocal tract models used in this research are a subset of the collection of STL files produced from MRI scans as part of Speed et al. [88]. The MRI scan produces a point cloud which represents areas of solid matter within the scanned region. 'Seed points' are placed within the empty space left in the tract and those points are then expanded out like balloons until they fill up the empty space and combine into one solid object which has the same shape as the airway of the tract. These solids are then exported and used to define the inner surface of the vocal tract. Two STL files produced from these surfaces can be found in Figure 4.1. These models were produced from imaging with a 2 mm resolution but were then oversampled to a resolution of $0.75 \times 0.75 \times 1$ mm. The validity of this oversampling is not known here however as this research is based on model geometry from this data set, it will be taken as accurate for these purposes. Models produced using these processes

frequently have an extremely high vertex count of around 500 000 and also tend to have a lot of anomalous geometry, like intersecting walls, and faces enclosed entirely within solid objects.

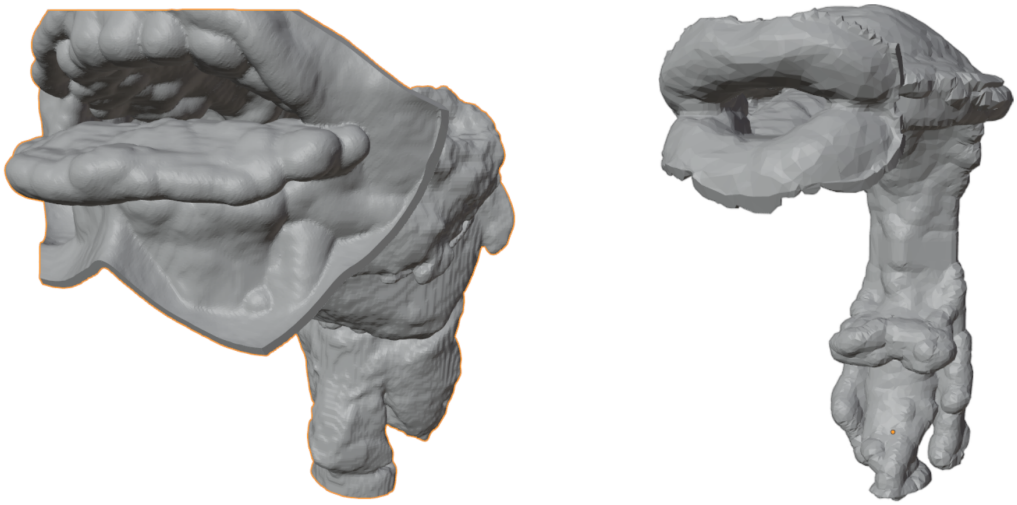


Figure 4.1: Left: Vocal tract model of Nesyamun 'True of Voice' [8]. Right: vocal tract model in the articulation of the vowel sound found in the word 'Stern'.

Models with such high levels of detail are cumbersome to manipulate during pre-processing, and would likely require a very high resolution simulation routine which would require a large amount of memory and a simulation time on the order of days if not longer depending on the machine. Simplification of these models is required to be able to conveniently use them without requiring these large memory allocations, rendering times in 3D modelling software, and simulation times. All the 3D modelling work done during this project so far has been done using the Blender software suite, which features many tools for simplifying and editing 3D geometry. The main tools used here are of geometry clean-up and of individual vertex manipulation.

The geometry clean-up tools in Blender can be used to reduce the number of vertices in a model in a variety of ways including dissolving faces and edges below a certain size, merging nearby vertices, and deleting loose and unphysical geometry. An example of using these tools to simplify models can be found in Figure 4.2. Using these tools, the models can be simplified in number of vertices by a factor of 100 without a complete loss of detail. The average displacement of the simplified

surface compared to the original surface of the model is on the order of 0.1 mm which is likely far below the eventual resolution of the simulation routine which will be used. This is not to say that these simplifications do not have a noticeable effect on acoustic output, and this may need to be investigated in the future. Figure 4.3 shows a zoomed in comparison between the original model and one which has simplified using Blender's geometry clean up tools. The model on the right has 136 times less vertices than the one on the right. While variation is clearly visible, the general shape remains recognisable. A simplification to this extent is likely greater than that which is needed for simulation, and is shown as a point of comparison.

Figure 4.2: Vocal tract model of Nesyamun 'True of Voice' with decreasing number of vertices from 500 000 on the left to 3000 on the right.

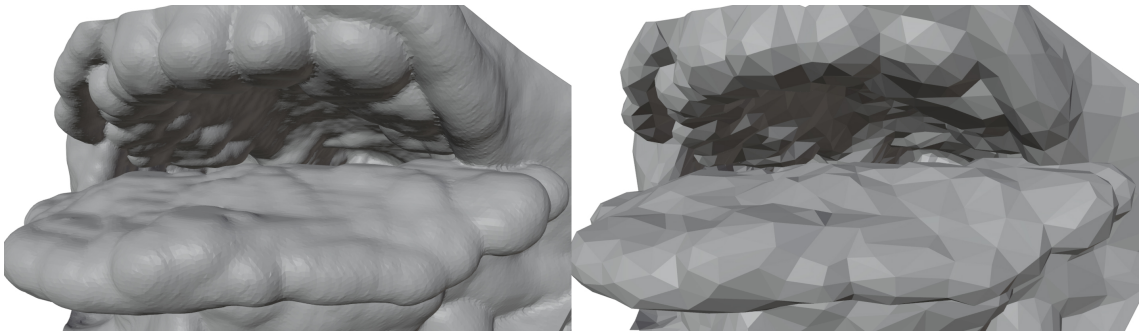


Figure 4.3: Vocal tract model of Nesyamun 'True of Voice'. The model on the right has been reduced in vertex count by a factor of 136. Variation is clearly visible but the general shape is preserved.

The automatic clean-up tools present in Blender are extremely powerful, but often are not capable of resolving particular complex geometric errors. The ability to edit models on an individual vertex basis allows for very precise adjustments to

the model and for control over the geometry that may prove difficult using other techniques. Figure 4.4 shows the ability to easily remove parts of the model to investigate different acoustic systems within the vocal tract.

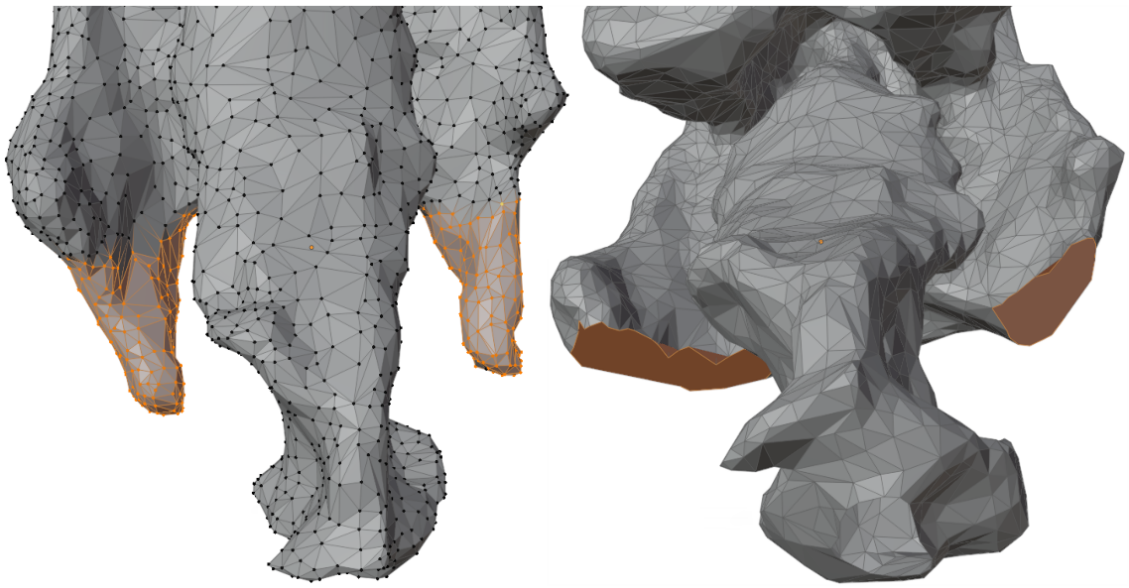


Figure 4.4: 'Stern' vocal tract model showing the ability to Select multiple vertices using visual tools and cut off sections of the geometry by deleting vertices and re-capping geometry. In the right image, the vertices highlighted in the left image have been removed and the holes left have been capped. The model has been rotated to more clearly show this.

Once a model has been simplified for ease of use, many new processes become available for digitising and visualising vocal tract models. The most important of these processes to the work on acoustic simulation is that of voxelisation.

4.1.2 Voxelisation

Voxelisation is the process of taking an arbitrary 3D model and then converting it into a grid of regular voxels, or identical 3D shapes. In essence, the MRI scans used to make the 3D models are also a voxelisation process: a 3D MRI scan is produced by taking a number of 2D images of the subject separated by a fixed distance and then using the colour of the pixels which comprise those images to define isometric cubes of material. The method described for MRI scans is fundamentally the same as the process used to voxelise any 3D model directly.

To voxelise a 3D model, it is first partitioned into a number of horizontal slices of unit thickness. A binary mask is then applied to each slice across a 2D Cartesian grid which is overlaid. This binary mask is set to true whenever the cell in the grid is considered occupied, usually dependent on how much of that cell is occupied by material, and false in the opposite case. For simplicity, this process is applied to each slice once they have been projected onto a 2D plane. Once the binary mask has been calculated for each slice, the slices are then stacked back on top of each other and wherever a 3D cell is bounded by two filled 2D cells it is filled with a voxel. When using cubic voxels, this produces a 3D model like the one shown in Figure 4.5. This approach of defining voxels is potentially error-prone, for example in a situation where there was a gap in the model that is on the order of the height of one voxel, the surface above and below that gap would cause that gap to be considered filled. These errors are minimised by maximising voxel resolution, with the voxelised model tending towards the original model as resolution tends to infinity.

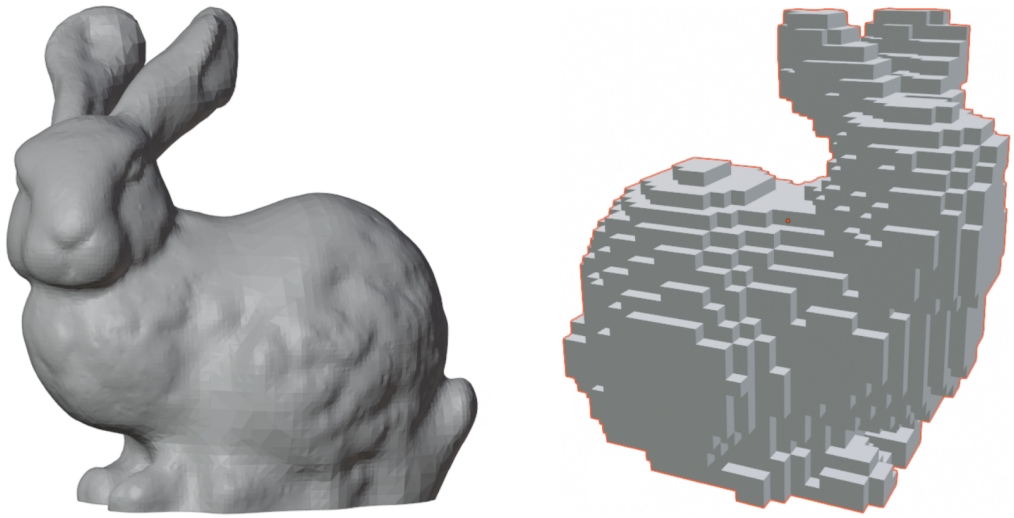


Figure 4.5: The Stanford bunny test model and its corresponding voxelised counterpart.

These voxelised models keep an approximation of the original geometry while representing it as an $L \times M \times N$ array of 1s and 0s corresponding to the location of the solid geometry, which is very simple for use in computational processes. While the obvious application is in providing a 1:1 interpretation of any input geometry into a Cartesian grid for an algorithmic acoustic simulation, there are also many

emergent methods of converting this model into other mediums that are designed to be composed of individual, or connected, voxels.

A greedy merging algorithm can be performed on the individual voxels, which attempts to merge them into one of a subset of predefined multi-voxels, with a focus on creating the maximum number of connections with vertical neighbours. By setting the predefined larger components to be the same size as various Lego bricks, a voxelised model can be converted into a valid Lego model [90]. Once the algorithm finds no more voxels to merge into valid shapes, it searches for instances of multiple connected components that are only supported by a single block. Blocks in the area of the supporting block get split back into individual voxels and then re-merged to improve the physical strength and connectivity of the Lego structure. This technique was used on the Nesyamun vocal tract model to produce a Lego recreation of the vocal tract of the 3000-year-old mummy, which can be seen in Figure 4.6. This Lego model is very accessible to a mainstream audience due to the familiar and recognisable medium, and has clear value in teaching and outreach.

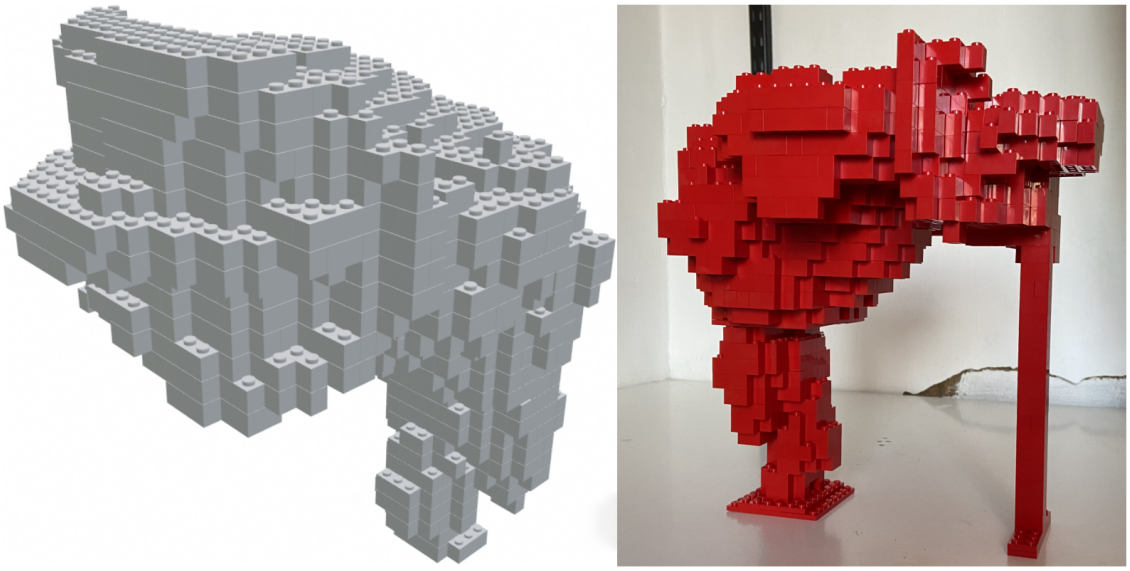


Figure 4.6: Left: The voxelised model of the vocal tract of Nesyamun 'True of Voice' merged into valid Lego bricks. Right: The Lego model constructed from standard Lego bricks.

Another familiar and recognisable medium to a mainstream audience is the video game Minecraft. Objects in Minecraft are constructed using cubic voxels, and so a

conversion between the voxel positions and a file readable by Minecraft is simple. The vocal tract of Nesyamun 'True of Voice', and of a tract articulated to produce the vowel sound in 'Stern', can be seen inside Minecraft in Figure 4.7. These models are increased in scale by a factor of 500, allowing a user to fly inside the tract and explore along its length. By editing game files, the sounds produced by both of these tracts can also be toggled on and off with an in-game switch. While this sound has no auralisation at all, it does still provide insight to a general audience that the sound made by the tract is dependent of the differences they see between the geometry of the two models.

Figure 4.7: Vocal tract models imported into the video game Minecraft. Left: The vocal tract of Nesyamun 'True of Voice' Right: A vocal tract articulated to produce the vowel sound in 'Stern'.

While voxelisation presents an extremely convenient way to import 3D geometries into simulation and software, the question of the effect of the inherent loss of detail with finite resolutions is an important one. As previously discussed, the acoustic frequencies with relevance to speech perception range from as low as a few hundred Hz and as high as 22 kHz [87]. A sound waves in the extremes of this range have wavelengths from 15 cm to 64 m. Measurements of the mid-sagittal axial length in 3D modelling software of some of the vocal tracts used in this work give a

length range of 0.235 m to 0.3 m. Preliminary investigations of appropriate voxelisation resolutions for use with vocal tracts returns resolutions in the range of 1 mm to 2 mm. Also as previously discussed, the size of the vocal tract means that acoustic propagation within it will be firmly with in the modal, or wave-based, regime. Specifically, standing plane waves will form between the glottis and the lips with a proportion of acoustic power radiating out of the tract at the lips. The treatment of this system, described in Section 2.1, as a series of connected tubes with varying radii as such should be valid.

The relevance of 3D geometry in this system then is in specifically defining the size of these tubes and how they connect together in the case of side passages. The relevance of voxelisation to this is that minor errors in wall position will change the effective radius of that section, which will effect that section's resonances. This minor error could occur many times along the length of the tract and the sum of those minor variations could lead to a large error in the output. Characterising the effect of these minor changes is complicated and could be done by adding and removing individual voxels along the tract between simulations. However, the nature of the voxelisation approximation can present an argument for these errors being nullified across the entire tract. As voxelisation as a procedure is unbiased as to whether it marks a position as filled or empty, the probability of and effect of over-estimations should be equal to that of under-estimations. These contributions, as such, should cancel out however in practice this will not always be exact. Still, the un-cancelled error should still be small.

4.2 Acoustic Simulation Development

As discussed at the beginning of this chapter, it has been deemed necessary to create a bespoke simulation package that was ideally suited for the needs of this research. To restate those needs, the simulation needs to produce accurate output throughout the tract and the simulation needs to be useable without the requirement for a great deal of technical knowledge in the field of acoustic simulation. There are

several secondary needs, like run times on the order of hours not days and ease of visualisation and analysis, but these are not as relevant to preliminary development.

The Python programming language has many desirable characteristics for the purpose of initial design. While a compiled language would produce an executable file which is generally understood by an average user, it would greatly slow down the development and prototyping process, and would also prevent knowledgable users from easily making changes to the functionality of the program. Python is also very frequently used in the scientific world due to its open-source nature and its large and accessible library of community created modules. An interpreted language such as Python does create two main issues for the final version of this package: a user would need to install the correct version of Python themselves and then run a script that will fetch any required dependencies, and interpreted languages require significantly more work to perform with comparable speeds to compiled languages, if even possible for some procedures. The final version of this simulation routine could be translated into a faster language if deemed valuable, but at this time Python is the language of choice.

The first step to producing a simulation of acoustic propagation in the vocal tract is deciding which method will be used for computing the acoustic propagation. The two algorithms tested here are the Finite-Difference Time-Domain (FDTD) and Digital Waveguide Mesh (DWM) methods. These methods were chosen initially due to their simplicity of implementation in the 3D Cartesian space provided by a voxelised 3D model. Other wave-based methods are more sophisticated and may produce more accurate results however, as this research is more focussed on the application of the simulation, their use was not deemed necessary.

4.2.1 Finite-Difference Time-Domain Method in Python

The foundational theory of this method has been discussed in Section 2.4.1. In this section some details of the implementation itself, as well as some additions to improve accuracy, will be discussed. The implementation here is largely based on

information from previous works, but the fundamentals underpinning the method are the same for all applications [91, 9, 92, 93].

The update Equations 2.13 and 2.14 are at this point fairly outdated. Newer formulations such as in Kowalczyk and Van Walstijn [91, 9] remove the need to update both particle velocity and acoustic pressure. By approximating the time and space derivatives present in the acoustic wave equation,

$$\frac{\delta^2 p}{\delta t^2} = c^2 \nabla^2 p, \tag{4.1}$$

with finite difference operators, equations for the time and spatial derivatives can be derived. The derived equation for the time derivative is given as

$$\frac{\delta^2 p}{\delta t^2} = \frac{p^{n+1}_{l,m,n} - 2p^{n}_{l,m,n} + p^{n-1}_{l,m,n}}{T^2} + O(T^2), \tag{4.2}$$

with T as the time step and $p^n_{l,m,n}$ as the acoustic pressure, and for the spatial derivative as

$$\frac{\delta^2 p}{\delta i^2} = \frac{p^{n}_{i+1} - 2p^{n}_{i} + p^{n}_{i-1}}{I^2} + O(I^2), \tag{4.3}$$

where i is the spatial dimension x, y, or z, p^n_i is the acoustic pressure holding the indices of dimensions not corresponding to i constant, and I is the grid spacing in the current dimension. Assuming that grid spacing and time step are constant, the acoustic wave equation becomes

$$\begin{aligned} p^{n+1}_{l,m,n} =& \lambda^2 (p^n_{l+1,m,n} + p^n_{l-1,m,n} + p^n_{l,m+1,n} + p^n_{l,m-1,n} + p^n_{l,m,n+1} + p^n_{l,m,n-1}) \\ &+ 2(1 - 2\lambda^2) p^n_{l,m,n} - p^{n-1}_{l,m,n}, \end{aligned} \tag{4.4}$$

where λ, the Courant number, equals cT/I. This can be more simply stated as

$$p^{n+1}_{l,m,n} = \lambda^2 \left[\sum_{adjs} p^n_{adj} \right] + 2(1 - 2\lambda^2) p^n_{l,m,n} - p^{n-1}_{l,m,n}, \tag{4.5}$$

where p^n_{adj} represents the acoustic pressure at an adjacent point within all adjacent points, labelled '$adjs$'. This equation can be used at any point in free space to calculate pressure at the next time step. Most of the work required in writing this simulation came from properly handling boundaries in the simulation. If only the free space equations are used and elements too close to the edge of the simulation domain are not included in the algorithm, then the edges act as fully reflective acoustic surfaces which is likely non-physical in this circumstance.

The method for accounting for surfaces in the simulation domain here is that of Locally Reacting Surfaces, where the normal component of the particle velocity at the point on the surface of the boundary depends only on the pressure at the point in front of the boundary and not the pressure at the points in front of the neighbouring boundary points [91, 9]. When formulating the LRS, its update equation has a term proportional to the value of pressure at the point past the surface in the boundary of the simulation media. These points are referred to in the literature as 'ghost points', as they are outside of the simulation domain and don't exist from the perspective of the acoustic propagation, and provide a useful way of thinking about an efficient approach to simplifying the FDTD algorithm at boundaries.

It can be shown that in a standard rectilinear Cartesian grid, each point is either within the domain or on one of three types of domain boundaries: walls, outer corners, and inner corners. To define these domain boundary types, it is most convenient to consider the number of ghost points adjacent to the point on the domain boundary in the 2D case of the FDTD algorithm and then simply expand them into the 3D case.

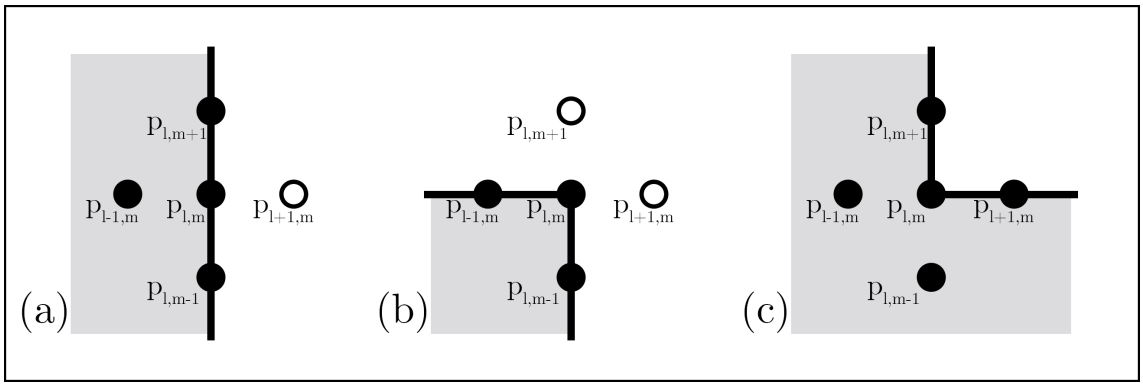


Figure 4.8: 2D standard rectilinear grid around domain boundaries. The grey region and black points represent the simulation domain and points within it respectively. The white region and white points represent the areas outside the domain and the ghost points respectively. The three domain boundary types shown are (a) a wall, (b) an outer corner, and (c) an inner corner [9].

Figure 4.8 shows the three types of boundary in a 2D standard rectilinear grid. For a point on a wall type boundary in 2D, two of its adjacent points will also be on the wall and one point will be opposite the wall. All three of these points are within the simulation domain, and the final adjacent point is outside the domain and is thus a ghost point. An outer corner has one adjacent point in each direction on domain boundaries and in the simulation domain, with the points opposite those outside of the domain. Finally, an inner corner has no adjacent points that are outside the domain, so no ghost points. Thus, any point in the simulation medium that is on a domain boundary either has one adjacent ghost point and is a wall, has two adjacent ghost points and is an outer corner, or has no adjacent ghost points and is an inner corner. As there is no ghost points in the inner corner case and there is no actual interaction with the boundary and only parallel to it, inner corner nodes can be updated as if they were in free space. In 3D, a wall still has one adjacent ghost point and inner corners still have no adjacent ghost points. There is however two types of outer corner when moving to 3D, the 'enclosed' outer corner with three adjacent ghost points and the 'open' outer corner with only two. A visual representation of these four boundary types can be seen in Figure 4.9.

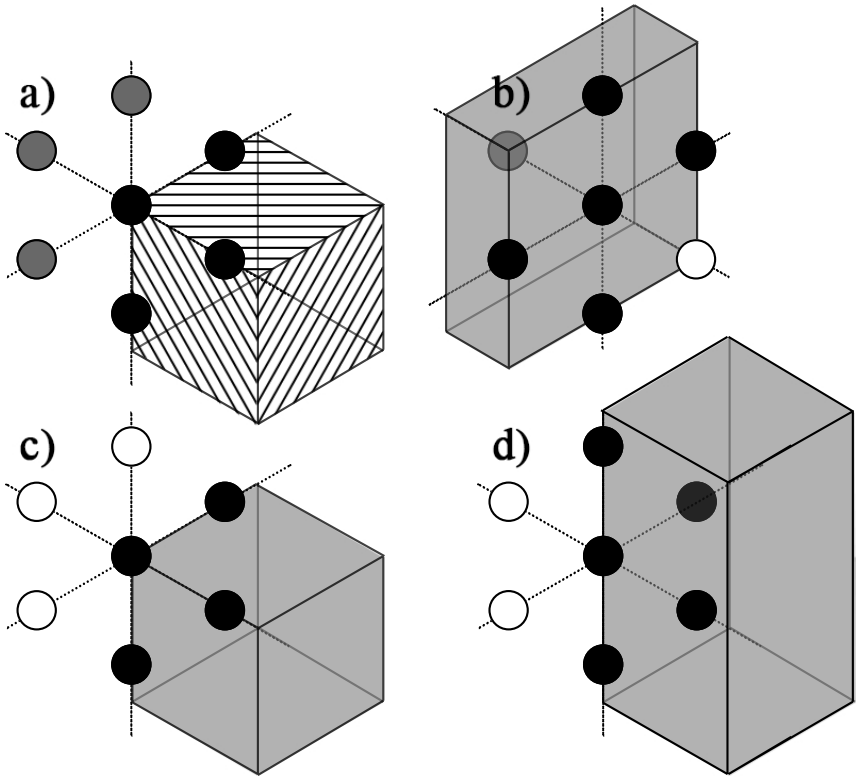


Figure 4.9: 3D standard rectilinear grid around domain boundaries. Grey regions show the simulation domain, with black points being on a domain boundary and grey points being fully within the domain. The white regions and white points represent the areas outside the domain and the ghost points respectively. The four domain boundary types shown are a) an inner corner, b) a wall, c) an 'open' outer corner, and d) an 'enclosed' outer corner. In the case of the inner corner the hashed area shows the are outside the domain with the rest of the region being within the simulation domain, for visual clarity.

The formulation of the boundary update equations begins with approximating the time and space derivatives present in the boundary condition equation, derived from the conservation of momentum

$$\frac{\delta p}{\delta t} = -c\xi_w \frac{\delta p}{\delta x}, \tag{4.6}$$

where $\xi_w = Z_w/\rho c$ is the specific acoustic impedance of the boundary. As in Equations 4.2 and 4.3, the centred finite difference operators are used for this approximation. Following the same procedure as earlier produces an update equation for the pressure at the ghost point:

$$p^n_{l+1,m,n} = p^n_{l-1,m,n} + \frac{1}{\lambda\xi_w}\left(p^{n-1}_{l,m,n} - p^{n+1}_{l,m,n}\right). \tag{4.7}$$

For points adjacent to ghost points, the ghost point update equation is substituted in to cancel out the ghost point. For a wall (ghost point in the $+x$ direction) the update equation becomes

$$\begin{aligned} p^{n+1}_{l,m,n} = \big[\lambda^2 \left(2p^n_{l-1,m,n} + p^n_{l,m+1,n} + p^n_{l,m-1,n} + p^n_{l,m,n+1} + p^n_{l,m,n-1}\right) \\ + 2\left(1-2\lambda^2\right)p^n_{l,m,n} + \left(\frac{\lambda}{\xi_x} - 1\right)p^{n-1}_{l,m,n}\big] \Bigg/ \left(1 + \frac{\lambda}{\xi_x}\right) \end{aligned} \tag{4.8}$$

and the update equation for an 'enclosed' outer corner becomes

$$\begin{aligned} p^{n+1}_{l,m,n} = \big[2\lambda^2 \left(p^n_{l-1,m,n} + p^n_{l,m-1,n} + p^n_{l,m,n-1}\right) + 2\left(1-2\lambda^2\right)p^n_{l,m,n} \\ + \left(\frac{\lambda}{\xi_x} + \frac{\lambda}{\xi_y} + \frac{\lambda}{\xi_z} - 1\right)p^{n-1}_{l,m,n}\big] \Bigg/ \left(1 + \frac{\lambda}{\xi_x} + \frac{\lambda}{\xi_y} + \frac{\lambda}{\xi_z}\right). \end{aligned} \tag{4.9}$$

These update equations, again presented by Kowalczyk and Van Walstijn [91, 9], have been reformulated in this work into a single general form of the boundary update equation for any boundary:

$$\begin{aligned} p^{n+1}_{l,m,n} = \big[\lambda^2 \left(\sum_{real} p^n_{real} + \sum_{ghost} p^n_{real}\right) + 2(1-2\lambda^2)p^n_{l,m,n} \\ + \left(\sum_{ghost} \frac{\lambda}{\xi_i} - 1\right)p^{n-1}_{l,m,n}\big] \Bigg/ \left(1 + \sum_{ghost} \frac{\lambda}{\xi_i}\right), \end{aligned} \tag{4.10}$$

where the sum over ghost points adds the point on the opposite side of the current point, and ξ_i is the specific acoustic impedance. Equations 4.5 and 4.10 as such provide a full description of the update equation for an FDTD scheme, notably not including frequency dependent effects. The form of equation 4.10, which is a general form for the boundary equations, has not been found to have been derived from the specific boundary equations in any previous works during this research. While this form is entirely equivalent to the other forms in terms of robustness and stability, it may allow for some conditional statements to be removed which would slightly reduce algorithmic complexity.

FDTD has a simple implementation from a memory allocation standpoint, requiring only a $3 \times L \times M \times N$ array to store the values of pressure at each node for the previous, current, and future time step, and a $L \times M \times N$ array to store the geometry in terms of whether each point is in free space or not. Running through the algorithm then only requires iterating over all the points in the domain boundary, checking if they are adjacent to any ghost points, and then applying the relevant update equation. Once every point in the domain has been updated, the algorithm steps forward in time and starts again.

All of the above was implemented in Python and used to perform acoustic simulations of propagation in the vocal tract. This method unfortunately had consistent issues in both implementation and output. While implementation of the algorithm itself was simple as expected, the implementation of LRS frequently led to pressure increasing at nodes within tightly confined space and never allowing that pressure to dissipate despite there being no barrier to that dissipation. In terms of outputs a series of errors were experienced in the form of inaccuracies caused by the lack of absorbing boundary conditions, and as pressures tending to infinity due to improper implementation of the pressure source. While both of these issues could likely be resolved with further development and sophistication in the simulation algorithm, it was instead decided that switching to the similar DWM method would be more sensible due to the advantages of the method.

4.2.2 Digital Waveguide Mesh Method in Python

The Digital Waveguide Mesh (DWM) method, previously discussed in Section 2.4.2, is described by the update equations 2.15, 2.16, and 2.17 [56]. Despite being equivalent to FDTD, DWM was chosen as a method to further explore due to its treatment of boundaries within the simulation domain. DWM adds a parameter that describes the impedance to the acoustic propagation through the simulation domain: the Acoustic Admittance, Y. This parameter describes the geometry of the input model fully without needing to treat the edges of the airway as domain boundaries,

instead allowing for acoustic propagation within the walls of the tract with a lower admittance. As such, no LRS formulation is required except for at the edges of the simulation domain which should prevent any erroneous behaviour of the propagation caused by tight constrictions and their interactions with the update algorithm. To make use of this only requires the storage of admittance in every waveguide, irrespective of flow direction.

On the subject of storage and memory usage, the implementation of this algorithm is significantly more complicated than that of the FDTD method. The scheme requires the storage of pressure in each scattering junction, incoming and outgoing pressures to and from each junction into all six of their surrounding waveguides, and values of the acoustic admittance in every waveguide. The junction pressure is still stored as a $3 \times L \times M \times N$ array containing the values of pressure at each node for the previous, current, and future time step. Each node in the simulation has six admittances associated with it, stored in another $3 \times L \times M \times N$ array. In this array, the three components represent the admittances in the L, M, and N direction separately with the index $[l, m, n]$ representing the waveguide between the current node and the previous node in that direction, and the index $[l+1, m, n]$ representing the waveguide between the current node and the next node in that direction. This method of 'forward-backward' indexing is also used in the arrays storing the waveguide pressure flow, which is the most complicated of the three memory allocations. The guide pressures are stored in three $2 \times L \times M \times N \times 2$ arrays for x-axis, y-axis, and z-axis waveguides respectively. The first two components store the pressure in each guide at the current time step and the next time step. The forward-backward indexing is used to index the different waveguides as in the admittance matrix, but with the addition of each waveguide storing two values for pressure: the pressure flowing in the negative direction to the previous junction, and the pressure flowing in the positive direction to the next junction. As an example, pressure flowing from the previous junction in the L direction to the current junction at the current time step would be indexed as $[1, l, m, n, 1]$.

With admittance dealing with boundaries internal to the domain, the only remaining boundaries to deal with are those at the edges of the domain. As DWM is equivalent to FDTD, a similar formulation to that of the FDTD boundary conditions can be used and applied only to the edges of the simulation. The general form of this boundary update equation, based on the same combination of equations which produced Equation 4.10, is

$$p_{l,m,n}^{n+1} = \lambda^2 \left(\sum_{real} p_{real}^{n} + \sum_{ghost} p_{real}^{n} \right) + \left(\frac{G_{sum} - \sqrt{3}}{G_{sum} + \sqrt{3}} \right) p_{l,m,n}^{n-1}, \tag{4.11}$$

where $p_{l,m,n}^{n}$ is the acoustic pressure at junction $[l, m, n]$ and time step n, p_{real}^{n} is the acoustic pressure at any real adjacent junction, and $\lambda^2 = \sqrt{3}/3(\sqrt{3} + G_{sum})$ in terms of the $G_{sum} = \sum_{ghost} G_{ghost}$ or the sum of the normalised admittances G between the current node and adjacent ghost points. G_{ghost} represents the acoustic admittance from the simulation region to the 'far field' and is initially set to 1 in these simulations in an attempt to minimise reflection back into the simulation region. Notably, this DWM version of the general LRS update equation is missing the term dependent on $p_{l,m,n}^{n}$. This general form is a combination of the LRS update equations derived specifically for DWM, which all lack the $p_{l,m,n}^{n}$ term, and will be used going forward. This general form of the LRS update equation is a unique output of this research, but is equivalent to the explicit equations, only providing benefit in simplicity and potential algorithmic complexity improvements based on implementation.

Having fully described the simulation procedure through the acoustic admittance, the 'free space' update equation, and the simulation boundary update equation, this algorithm is now general to any simulation domain. At each time step, simply check how many adjacent nodes are ghost points, and then apply the corresponding update equation.

Notably, the implementation described here is not the only possible implementation of a digital waveguide mesh. Two major options for DWM simulation exist: Wave Based-Digital Waveguide Mesh (W-DWM), and Kirchoff Type-Digital Waveg-

uide Mesh (K-DWM) [57]. The simulation method described in this thesis so far is the W-DWM scheme. From first principles, the formulation of the DWM update equations described here are based on the d'Alembert solution of left and right-travelling wave components that sum to the whole waveform. This produces the need to track incoming and outgoing pressure at each waveguide. The K-DWM is directly equivalent to FDTD modelling in that instead of applying a specific solution to the wave equation, it instead approximates the partial derivatives with a centred difference in time and space. Note that FDTD, K-DWM, and W-DWM are all equivalent but with implementation differences, and the consideration of the differences between implementations of K-DWM and W-DWM is here used as a convenient comparison between the advantages and disadvantages of both methods.

The main advantage of W-DWM over K-DWM is that the solution is, in theory, exact due to its foundation on the d'Alembert solution. K-DWM is only approximate and so can not always produce truly accurate results. K-DWM also encounters numerical instability at high frequencies. In reality, careful consideration needs to be made to exploit the exactness of W-DWM. For example, if W-DWM simulations are not run on fixed-point values and instead use floating-point arithmetic then instability is introduced due to limitations on binary representations of decimal numbers. On the other hand, K-DWM requires significantly less memory than W-DWM. FDTD, which is closely related to K-DWM, required a total of $4 \times L \times M \times N$ memory positions in the implementation described in Section 2.4.1, whereas W-DWM requires $10 \times L \times M \times N$ memory positions in this implementation. Despite this memory requirement, W-DWM will be chosen as the main simulation method for this research due to its inherent accuracy and its more prevalent use in similar research.

4.3 Simulating Acoustic Propagation in the Vocal Tract Using W-DWM

Having written and implemented a DWM algorithm for solving the acoustic propagation in any general domain, this method was then applied to the vocal tract. The first step of performing this simulation is to voxelise the input model. Voxelisation here is achieved using a publicly available online tool [94]. A custom script was written that takes a file containing the positions of every voxel from the voxelised model of the vocal tract and uses those positions to populate the acoustic admittance arrays. The acoustic admittances are defined as the inverse of the acoustic impedance, Z, which is itself defined as the product of the density of the medium, ρ, and the speed of sound in the medium, c. For propagation through air, a density of $1.14\,\mathrm{kg\,m^{-3}}$ and a sound speed of $350\,\mathrm{m\,s^{-1}}$ were chosen based on works by Arnela et al. [24, 95] for ease of comparison, however these values are estimates and may not be truly accurate to the real physics, which produces an acoustic impedance of $399\,\mathrm{Pa\,s\,m^{-1}}$. For propagation through tissue an acoustic impedance of $83\,666\,\mathrm{Pa\,s\,m^{-1}}$ was selected based on Švancara and Horáček [96] for ease of comparison. This value is specifically for the impedance through flesh, with Wang et al. [97] presenting a value of $1.5\,\mathrm{MPa\,s\,m^{-1}}$ as a resultant acoustic impedance of the head when also including bone and other materials. If the simulation domain here included more of the head that surrounds the tract then the higher impedance value is likely more accurate, however the simulation region here is small and will include much less bone and other materials, so a value for impedance closer to that of just flesh is likely optimal. The admittance array only needs to be calculated and saved once per geometry.

To determine the other parameters of the simulation, a careful consideration of the stability of the simulation is required. A 3D FDTD simulation is considered stable for values of the Courant number $\lambda = c_{air}\Delta T/\Delta X$ less than or equal to $1/\sqrt{3}$. This is the Courant convergence condition and is a standard result in the

numerical analysis of partial differential equation solvers. ΔX is set by the size of the grid that the vocal tract geometry is voxelised into. For initial investigations, a grid spacing of 2 mm was used to keep simulation times reasonable without losing too much detail of the tract or causing it to become closed or cut off, and to remain accurate to the scan resolution of the input models. Rearranging for the time step ΔT gives a value of approximately 3.36 µs, or a sample frequency of approximately 298 kHz. Using a larger grid spacing would allow for a longer time step and as such a lower simulation time, however would result in a loss of geometrical accuracy.

Figures 4.10 and 4.11 show an example of a simulation output of the junction pressure at nodes along the direction of pressure flow through the vocal tract corresponding to the English phoneme /ɜ/. The first plot shows the first 3.3 ms of pressure data, and the second plot shows the full 33 ms of data. In the first set of plots the initial pressure pulse takes less than 0.65 ms to travel through the tract, the amplitude of the pressure variation decreases through the tract, and the pressure variation reduces in magnitude with increasing time, all of which are expected behaviour and lend some credibility to the simulation accuracy. The second set of plots reveals an inaccuracy that can be seen in 3D animations of the pressure in the domain. At long times, the pressure variation dissipates and gives rise to a smooth parabolic upward trend in pressure. This is believed to be due either to numerical errors caused by imprecise floating-point arithmetic or the result of a phenomenon known as solution growth which is caused by the implementation of the volume velocity source, which adds acoustic pressure to the system, as a 'soft source'.

There are two simple methods for implementing a source excitation within the model: the hard source and the soft source [40]. The hard source will set the pressure value at a chosen input node or set of nodes to a fixed value at every time step based on a given excitation equation. This method is trivial to implement but causes scattering of any incident pressure on that node. This will lead to unphysical effects that may have an effect on the simulation output, especially for any nodes near to the input node. The soft source instead adds the pressure value produced by the

excitation equation to the chosen source node at every time step. This means that the source node will properly obey the update equations and not cause scattering. However, if there is any constant offset in the input signal, which could be caused by a calculation inaccuracy, solution growth will occur leading to an exponential growth in background pressure level for the entire course of the simulation. The addition of the offset to the source node at every time step effectively acts as an input signal with infinite length that constantly provides a volume velocity to the system. This is technically not an inaccuracy in the model, but instead represents a completely unphysical situation. This offset is difficult to remove and leads to an increased loss of precision in floating-point calculations as the small and physical pressure variations are dominated by the growth of the offset. Through care and sophisticated treatment of the source implementation, typically through using a source equation designed to counteract the effects that lead to solution growth, soft sources can still be used in DWM simulations. As the source node will be close to the edge of the domain, it was deemed that any additional scattering at that node will have a similar effect to the scattering at the domain edge and so will likely not have a large effect on simulation output. This is an avenue for future work however.

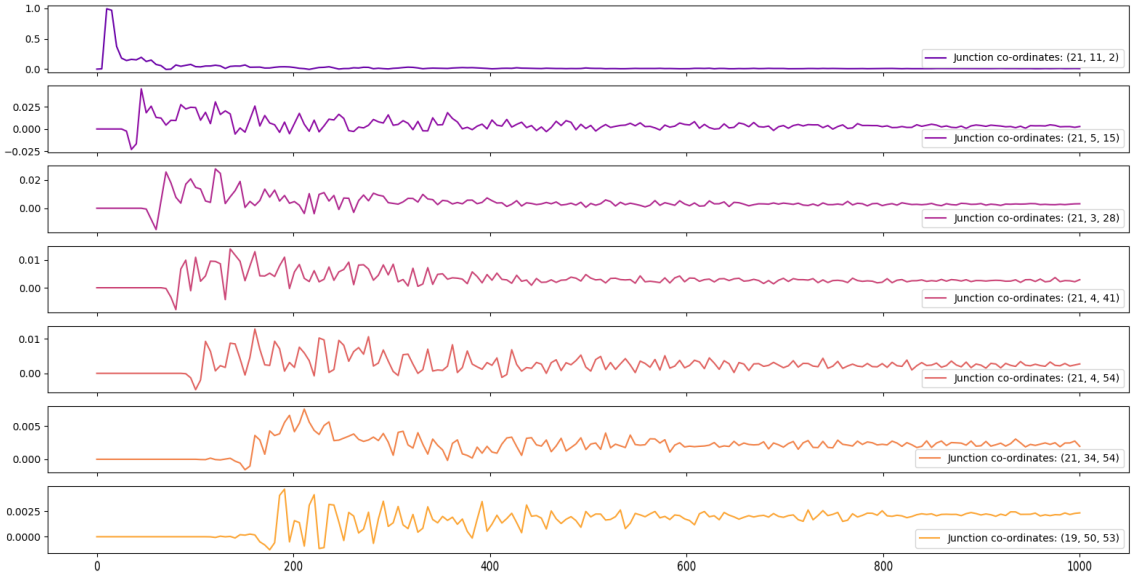


Figure 4.10: Plots of junction pressure from $0 < \mathrm{T}_{step} < 3.3\,\mathrm{ms}$. Y-axis scale is in arbitrary pressure units and x-axis scale is in time steps $\Delta t \approx 3.36\,\mu\mathrm{s}$. Plots arranged in order of flow through tract (top plot at source junction, bottom plot at lips).

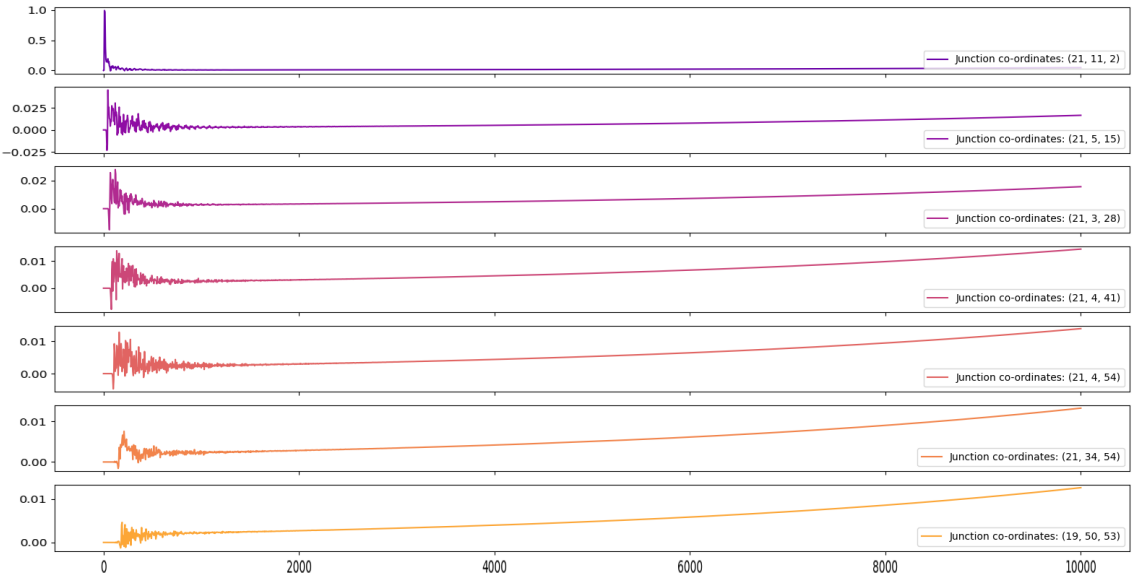


Figure 4.11: Plots of junction pressure from $0 < \mathrm{T}_{step} < 33\,\mathrm{ms}$. Y-axis scale is in arbitrary pressure units and x-axis scale is in time steps $\Delta t \approx 3.36\,\mu\mathrm{s}$. Plots arranged in order of flow through tract (top plot at source junction, bottom plot at lips).

Testing validity of the simulation against real values can be achieved using the peaks in the Vocal Tract Transfer Function (VTTF) produced at a listener location as in Equation 2.10. The sound output produced by the articulation of a particular vowel is driven by the filtering of an input source due to the resonances produced in the vocal tract. These resonances are called formants and can be seen in the

VTTF as peaks in amplitude. Comparing the frequencies of these peaks to formant frequencies from recorded speech can give a measure of simulation accuracy. Figures 4.12 and 4.13 show the VTTF for the previous 3.3 ms and 33 ms of pressure data respectively alongside the accepted formant values of this vowel sound from literature. While the 3.3 ms VTTF has a very low resolution, the values for the first three formants obtained from the first three peaks in the VTTF are slightly closer to the accepted values than those from the 33 ms data, likely due to not including the long time step errors discussed earlier. Many details not visible in the first plot can be seen in the second, especially above 4 kHz. The sampling frequency used for these simulations is 298 kHz due to the spatial resolution of the model. When a continuous signal is converted into a discrete one in time, that signal can experience aliasing effects at frequencies above half of the sample frequency, known as the Nyquist frequency. While the Nyquist frequency in these simulations would be 149 kHz which is far greater than the first five formant frequencies, prior research into the stability of various FDTD grid layouts puts the cut-off frequency for accuracy of the outputs of the Standard Rectilinear (SRL) grid used here as low as 0.196 times the sample frequency [98]. While this cut-off of approximately 58 kHz is still an order of magnitude larger than the highest formant frequencies this work is concerned with, there could still be other effects in the high frequency range that have an effect on this VTTF data. The significance of the errors in the formant frequencies in both VTTFs will be discussed in Section 4.3.1.

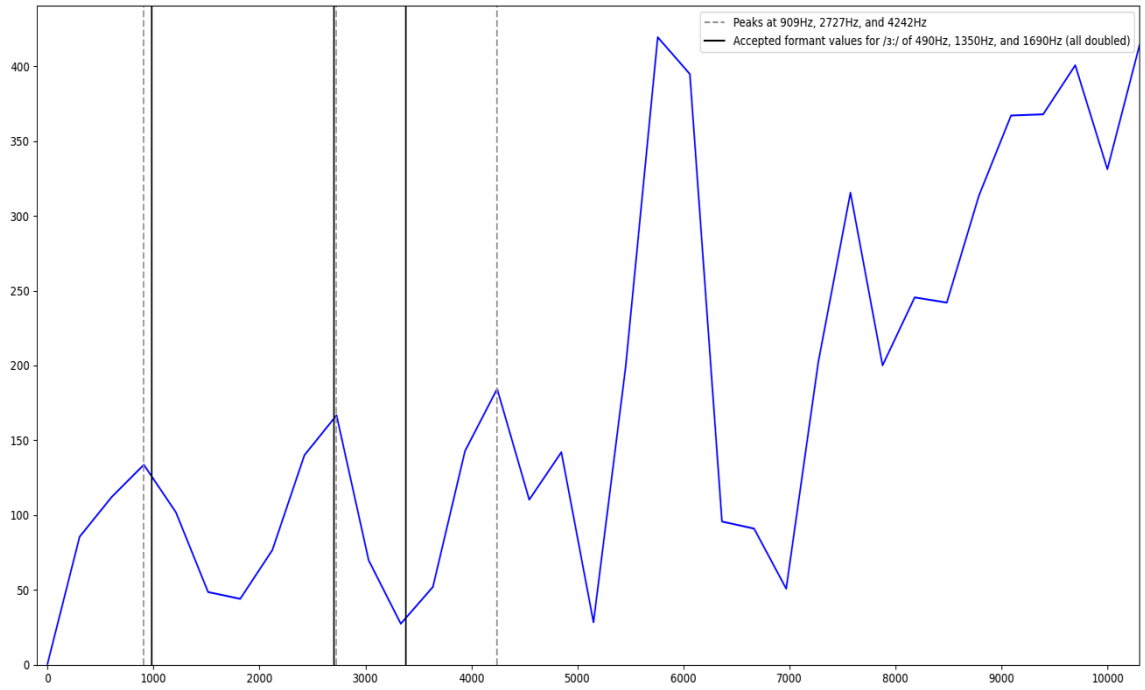


Figure 4.12: Vocal Tract Transfer Function (VTTF) produced using equation 2.10 for 3.3 ms of pressure data for 'Stern' model. Only the first 10 kHz is included for visual clarity. Spectral magnitude is on the vertical axis with arbitrary units and frequency is on the horizontal access in Hz. Frequency doubling of accepted formant values will be discussed in Section 4.3.1.

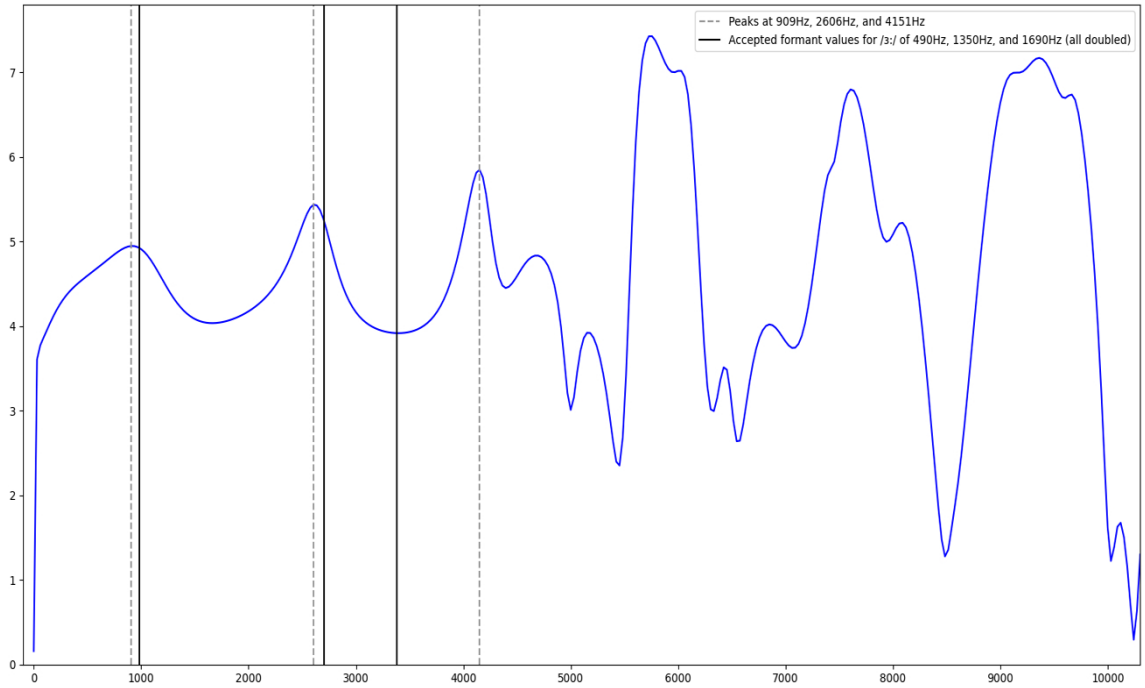


Figure 4.13: Vocal Tract Transfer Function (VTTF) produced using equation 2.10 for 33 ms of pressure data for 'Stern' model. Only the first 10 kHz is included for visual clarity. Spectral magnitude is on the vertical axis with arbitrary units and frequency is on the horizontal access in Hz. Frequency doubling of accepted formant values will be discussed in Section 4.3.1.

By employing a source-filter model as described in Section 2.2, the VTTFs produced here may be used for speech synthesis. By applying a VTTF as a filter to a source signal that encapsulates the voiced and unvoiced excitations at the base of the tract, a speech signal can be produced. Applying one dataset as a filter for the other simply requires multiplying them together as long as both data sets are in the frequency domain, which does add some computational complexity and some uncertainty from performing Fourier transformations. The source signal used to produce speech in this work so far is the Liljencrants-Fant (LF) model, which is a good approximation of the glottal waveform in the vocal tract [27]. Once filtered, the resulting dataset is transformed back in to the time domain and saved as the amplitude at each time step of an audio file format such as Waveform Audio File Format (WAV). Both the VTTF and WAV files can then be used in further analysis to extract formant values and investigate qualitative properties of the simulation output.

Having fully described the entire simulation process from STL pre-processing to formant measurement and speech sound production, this method can now be used on real vocal tract models and compared with recordings or prior research to assess accuracy.

4.3.1 Simulation Accuracy

Three vocal tract models have been used in the initial testing of this simulation. The first model is that of an adult male producing the sound /ɜː/ as in 'Stern'. The second model belongs to the 3000-year-old Egyptian mummy Nesyamun, True of Voice in the articulation that the body was recovered in. Finally, another vocal tract of an adult male, this time producing the sound /ɔː/ as in 'Port'.

'Stern' Model

Section 4.3 presented initial simulation data produced from the 'Stern' vocal tract model. As briefly discussed in that section, the peaks in this VTTF correspond to

the formants, or resonant frequencies, of the produced speech sound. As this model is producing a known vowel sound, the formants extracted from this plot can be compared to known formants to gauge the accuracy of the simulation. While this particular speaker's speech formants may vary from the literature, it is an appropriate initial test to ensure that the simulation produces outputs in an approximately correct range. Formants are often shortened for ease of discussion in literature, for example the first formant is usually abbreviated to F1, and that same format will be used here.

Table 4.1: Formant values for 'Stern' vocal tract model and simulation. Accepted formant values taken from Peterson and Barney [10].

Data set	F1 (Hz)	F2 (Hz)	F3 (Hz)
Accepted formant frequencies	490	1350	1690
Accepted formant frequencies (doubled)	980	2700	3380
VTTF peaks (first 3.3 ms)	909	2727	4247
VTTF peaks (33 ms)	909	2606	4151

By measuring the value of each formant as the frequency at which each peak is at its maximum, the formant values found in Table 4.1 are obtained. The accepted frequencies are significantly different to the values measured from the VTTF. If the accepted values are doubled, representing a shift by one octave upwards, the agreement between measured and given values is much better. For the full 33 ms of data, F1 would have an absolute error of 7.81 %, F2 an absolute error of 3.61 %, and F3 an absolute error of 18.57 %.

Notably, in speech, the first three formants are the most important for clarity of the vowel sound, with higher formant values governing the naturalness of the sound [10]. The relative frequency of the formants also plays a large part in vowel identification, not just the absolute frequencies. In the accepted values, F2 is 2.755 times larger than F1 and F3 is 1.252 times larger than F2. In the 33 ms VTTF data, F2 is 2.867 times larger than F1 and F3 is 1.593 times larger than F2. There is no

definition for how similar the relative frequencies must be to still be recognisable as the correct sound, but these relative frequencies are at least similar to the accepted ones.

While the source of this required doubling is unknown at this point and will be explored further in Section 4.3.2, if it is assumed that it is a result of simulation parameters and take it as truth, the simulation is observed to produce a reasonable estimation of F1 and F2 with higher frequency formants losing accuracy. This is expected to be caused by one of two possible effects.

The first would be the long timescale solution growth shown in Figure 4.11 causing errors over a range of frequencies. By only considering the first 3.3 ms of pressure data in an attempt to minimise the effect of these errors as in Figure 4.12, an increase in accuracy on F2 by 2.62 % and a decrease in accuracy on F3 by 1.84 % is observed. This is inconclusive and implies that if solution growth is a cause of these errors then removing the late time step data in this way is not a valid solution to it. The second possible cause of high frequency errors is in the geometry of the simulation itself. The method of LRS for treating domain boundaries, at least in the current implementation, does not allow the domain boundaries to act as a truly absorbing boundary, instead reflecting some pressure back in to the system. These unphysical reflections likely lead to similarly unphysical behaviour in the VTTF and are complex to remove, potentially requiring a more sophisticated boundary formulation. In Gully [4], a comparison can be found between three DWM simulations: a 2D W-DWM simulation, a 3D K-DWM simulation, and a 3D dynamic W-DWM simulation based on the same principles as this research. Values given in that paper are in terms of the mean percentage error on formant frequencies across six English monophthongs for formants up to the fifth. These values are much more robust and accurate than the values presented here for an initial simulation on a single vowel, but are used to provide a point of comparison. The average error on F1-F3 for the 'Stern' model in this work is 10.08 %. The work by Gully [4] presents a mean absolute error across F1-F5 of 14.01 %, but much of the observed accuracy comes from

a well reproduced F4 and F5. When only considering F1-F3, the mean absolute error increases to 20.28 % or 10 % less accurate than this work. This is promising, however the importance of F4 and F5, which this work so far poorly reproduces, should not be ignored.

By looking at spectrograms of the sound file produced by the simulation for this vocal tract with the spectrogram of another adult male English speaker producing the same vowel in Figure 4.14, it is observed that the produced frequency spectrum has much more obvious formants, which present as darker horizontal regions, all the way to 5 kHz whereas the simulation output only shows clear frequency variation up to approximately 1 kHz. This is suspected to arise during the filtering with the source signal and is discussed further in Section 4.3.2.

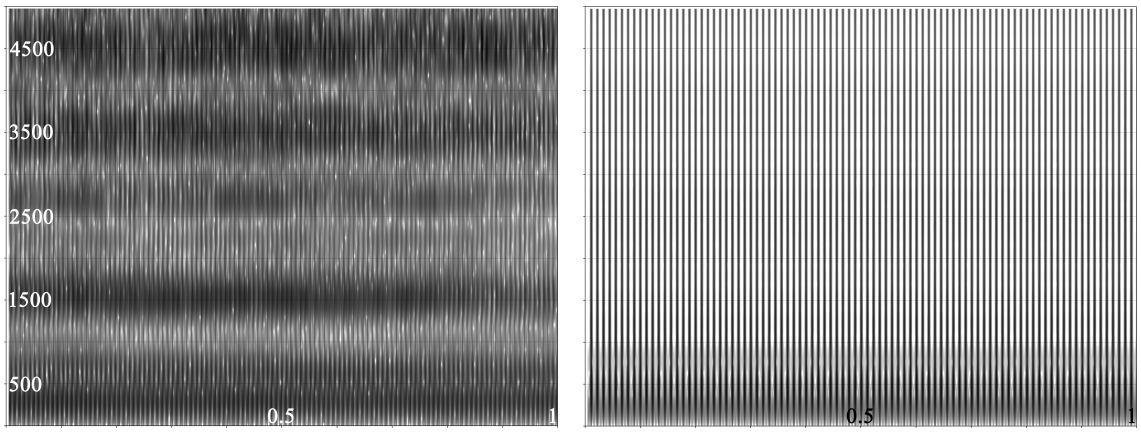


Figure 4.14: Spectrograms of a recorded /ɜː/ vowel production from Left: a male subject and Right: the output of the 'Stern' vocal tract model. Frequency on the vertical axis is in Hz, increasing in increments of 500 Hz from 0 Hz to 5000 Hz. Time in the sample file in on the horizontal axis in seconds, increasing in increments of 0.1 s from 0 s to 1 s.

Nesyamun, 'True of Voice' Model

A simulation was performed using vocal tract model geometry of the 3000-year-old mummy Nesyamun, 'True of Voice' as a point of interest due to previous works by this group [8]. The vocal tract of this subject is highly dessicated, lacking a tongue and not given in any particular articulation. This tract was however used to produce sound output using a physical vocal tract model and speaker, which could be used as a point of comparison with simulation results. Formant values were extracted

from a recording of the sound output from this physical model, using the formant estimation algorithm available in Praat which will be discussed further in Section 4.3.2, and are available in Table 4.2 [99].

Table 4.2: Formant values approximated from a recording of sound output from the physical vocal tract model of Nesyamun, 'True of Voice'.

Data set	F1 (Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)
Recorded audio	612	2074	2747	4034

The VTTF produced from this simulation can be found in Figure 4.15. By this point, the implementation of the volume velocity source in the simulation has been converted to a hard source and as such solution growth on the timescale of this simulation has been negated. The VTTF produced here looks very unusual compared to VTTFs obtained from living subjects. While this may be due to simulation errors, it may also be due to the preservation of the tract obscuring the formants. Frequency scaling as seen in the plot of the 'Stern' model has not been performed here.

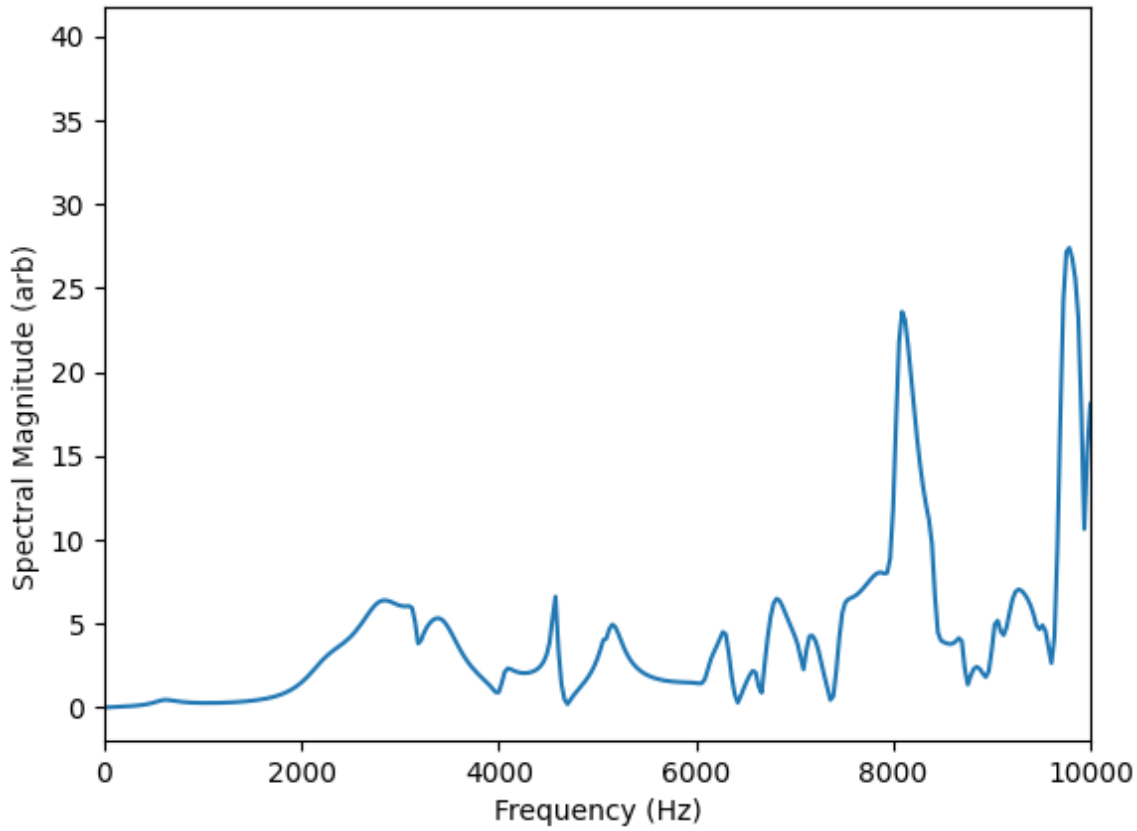


Figure 4.15: VTTF produced from DWM simulation of the Nesyamun vocal tract. Only the first 10 kHz is included for visual clarity.

'Port' Model

Figure 4.16 shows the VTTF produced for a model of an adult male vocal tract producing the vowel /ɔː/ as in 'Port'. The given formant frequencies, obtained in the same manner as for the 'Stern' model, are shown on the plot however F3 falls off the scale of the graph. While F1 and F2 look to show good agreement, as with the frequency scaling required for the 'Stern' data, the frequencies for this data have been scaled by a factor of five. This frequency scaling has been applied to the data set itself rather than the accepted formant frequencies, hence the small range of the plot. The apparent agreement of F1 and F2 also ignores the small peak at approximately 200 Hz and ignores the presence of numerous peaks between F2 and the expected value of F3. The multiplicative factor differing between this data and the 'Stern' data raises concern over the source of this factor, and whether it is actually a result of simulation parameters or if it instead is purely a coincidence.

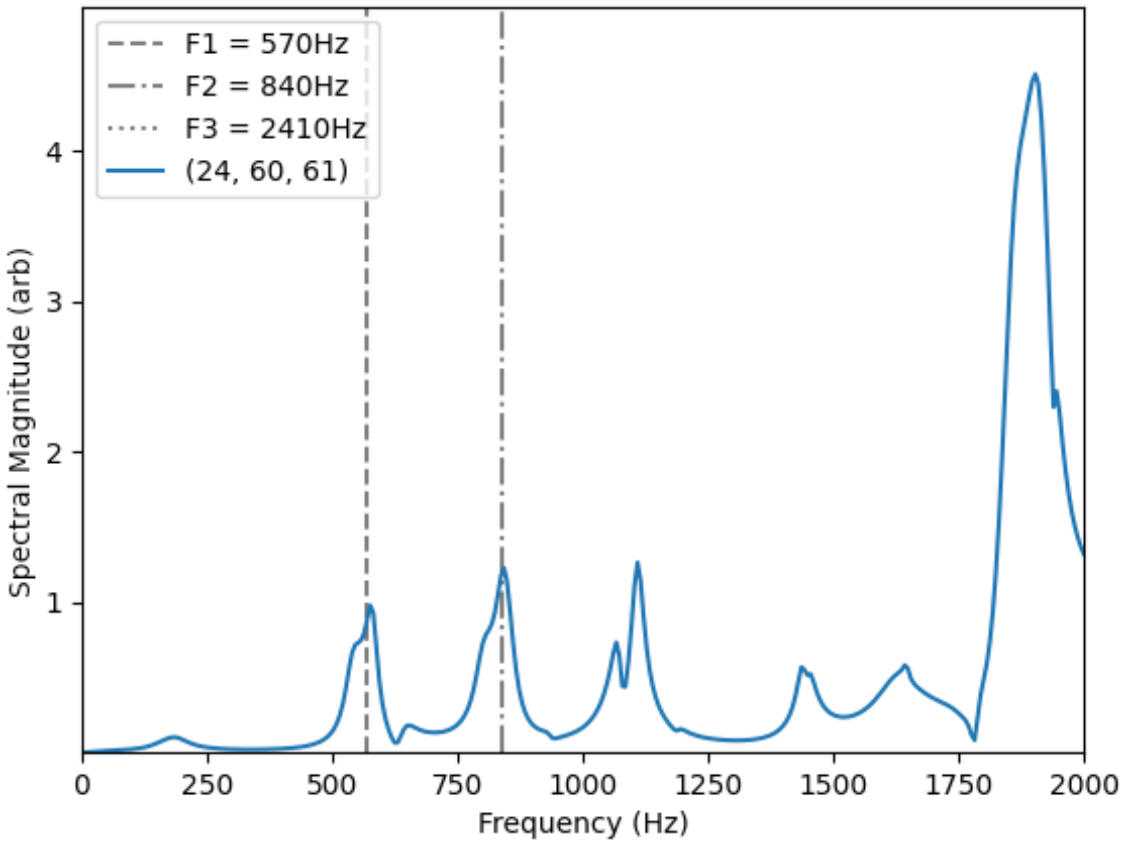


Figure 4.16: VTTF for 'port' vocal tract model. The first two formant values taken from Peterson and Barney [10] are plotted as vertical lines. The frequencies of the data set are scaled down by a factor of 5. Only the first 2 kHz is included for visual clarity. The dotted line showing F3 is as such not in the plotted range of this figure.

4.3.2 Improving Simulation Accuracy

The repeated issues arising from the use of the VTTF to investigate the accuracy of the simulation routine call the accuracy of the simulation into question. At this point to produce formant frequencies in an appropriate frequency range, the 'Stern' VTTF required the frequency values to be scaled down by 2, the 'Port' VTTF required the frequency values to be scaled down by 5, and the Nesyamun VTTF did not clearly produce any comparable values with a range of divisors, which will be discussed further in Section 4.4.4. The requirement of different dividing factors eliminates any simple solution which could be attributed to a spurious scaling factor in the code. Prior research which makes use of the VTTF has not mentioned experiencing this same issue.

One potential cause of this error is an inaccuracy in the routine used to perform

the Fourier transformation of the pressure data into the frequency domain. This work uses the Fast Fourier Transform (FFT) routine implemented as part of the Scipy package, which is itself an implementation of the Cooley-Tukey FFT algorithm [100]. This FFT algorithm is an exact computation of the discrete Fourier transform and theoretically has no errors. In real applications, floating-point arithmetic errors can lead to small errors in FFT outputs with an upper bound of $\mathcal{O}(\epsilon log N)$ where N is the number of data points, and ϵ is the relative precision of the floating-point arithmetic in use on the local machine [101]. This upper bound is likely quite small compared to the values outputted by the FFT, but it is in theory possible that FFT errors could lead to frequency scaling. A more likely source of error could be that speech production in the vocal tract cannot be well described as a Linear Time-Invariant (LTI) system. This Fourier transform implementation produces complex exponential functions which are only valid eigenfunctions for LTI systems. As the geometry of the simulation is explicitly static, the system should be necessarily time invariant. The glottis, here the source signal, is coupled to the vocal tract which can lead to non-linear behaviour in the output. Existing literature states that models such as this one should be adequately LTI, but it is possible that this particular vocal tract model fails on that front.

The fields of feedback systems and power systems frequently make use of the transfer function as a means of characterising the behaviour of a linear system. The mathematical rigour which is more common in those fields can potentially shed some light on the issues faced here with the multiplication of the frequency values. A given linear system may be described by the following equation

$$a_0\ddot{y}(t) + a_1\dot{y}(t) + a_2 y(t) = b_0\ddot{x}(t) + b_1\dot{x}(t) + b_2 x(t), \tag{4.12}$$

where a_n and b_n are the coefficients of the equation and y and x are the output and input respectively. Taking the Laplace transform of this equation produces the complex frequency domain equation for the system of

$$a_0 s^2 y(s) + a_1 s y(s) + a_2 y(s) = b_0 s^2 x(s) + b_1 s x(s) + b_0 x(s), \tag{4.13}$$

where s is the complex frequency. Taking the input and output variables out as factors on either side produces

$$\begin{gathered} \left(a_0 s^2 + a_1 s + a_2\right) y(s) = \left(b_0 s^2 + b_1 s + b_0\right) x(s) \textbf{ or} \\ a(s) \cdot y(s) = b(s) \cdot x(s), \end{gathered} \tag{4.14}$$

where $a(s)$ and $b(s)$ are the characteristic polynomials of the ordinary differential equations. Finally, the transfer function, $H(s)$, is redefined as the ratio of the output and the input:

$$H(s) = \frac{y(s)}{x(s)} = \frac{b(s)}{a(s)} = \frac{b_2 s^2 + b_1 s + b_0}{a_0 s^2 + a_1 s + a_2}. \tag{4.15}$$

Equation 4.15, adapted from Åström and Murray [102], allows for greater insight into the way the transfer function acts as a filter. The roots of the polynomial $b(s)$ are the complex frequencies at which the transfer function has a value of zero which will completely nullify the output in an ideal case. The roots of the polynomial $a(s)$ conversely are the poles of the transfer function, at which the output will have infinite spectral magnitude in the ideal case. It should be noted that, in real applications, impedance prevents either of these cases from reaching their unphysical ideals. The poles of the transfer function are the eigenvalues of the system itself, and as such the eigenvectors of the system can be scaled by any amount and still be valid. What this may mean in the context of the VTTF is that the VTTF could be scaled in frequency by any real scalar value and still be a valid solution to the eigenvalue equation $A\vec{v} = \lambda\vec{v}$ if the state variable of the system only concerns the frequency. This final point is key as the eigenvalue theory presented here is only strictly accurate if the state variable of the system does not include any other contributors, for example the spectral magnitude of the system. An investigation into this eigenvalue problem formulation of the vocal tract would be required to

confidently assert any of the above discussion, which is beyond the scope of this thesis. If it is indeed accurate, then this knowledge may still not lead to the easy removal of this scaling error.

When dealing with eigendecomposition, eigenvectors are typically normalised to avoid arbitrary scalar multiplications. If the above is correct, then the only way to remove the need for comparison between output data and recorded data would be to determine some algorithmic method of reducing the frequency domain back down to its 'unit-length', which would at least require knowledge of the expected formant frequencies. Again assuming the validity of the approach above, it is still troubling that no prior research has encountered this issue. This could either be due to the method other research uses to produce its VTTFs or an issue with implementation here. While other research reportedly uses the same method of generating VTTFs as described in Equation 2.10, it may be possible that the workflow they use which is frequently used within speech science doesn't do this explicitly. For example, if the Praat software package, commonly used in speech science, is already performing much of the data analysis then a function within Praat may produce the VTTFs and perform some pre-processing to de-scale the frequency domain.

What is potentially more likely is that the implementation of the simulation routine here doesn't prevent the appearance of this scaling in frequency. This simulation package has been largely developed in a way which is agnostic to its intended purpose. It may be that the more specifically designed simulation routines could set some limitations on the calculation of the VTTF that prevent this scaling. These limitations, if they exist, are not discussed in any prior research known of at this time.

Performing simulations to test these theories, optimise the simulation routine, and explore unexpected behaviours is currently quite difficult due to the complexity of the simulation domain. Simulations are being performed on real vocal tract models during this testing and so it is not always clear what effect any small change has had on the output and why, especially when familiarity with this simulation routine

is still low. Switching between models to test consistency of observed behaviours is also nearly impossible without much more knowledge on the quirks of the routine, with varying domain sizes, source locations, and geometry. To investigate more quantitatively the optimisation of each variable of the simulation, maximisation of the simplicity of the input, while also having a direct physical comparison to the real expected output, will be required.

Arai [11] proposed and produced a set of cylindrical models which have the same radius as the vocal tract along its length and are internally smooth which give accurate representations of five Japanese vowels, based on the work by Chiba and Kajiyama [103]. The 3D model files are available digitally for all five cylinders, and access to the physical set is also available. As such, audio recordings of the sound output from the physical models can be directly compared with simulation output to simplify the improvement of the simulation. The five cylinders can be seen in Figure 4.17. The VTTF produced from pressure data at the end of the tube corresponding to the Arai /a/ model is shown in Figure 4.18.

Figure 4.17: The five Arai vowel sound cylinders (from left: /i/, /e/, /a/, /o/, and /ɯ/) [11].

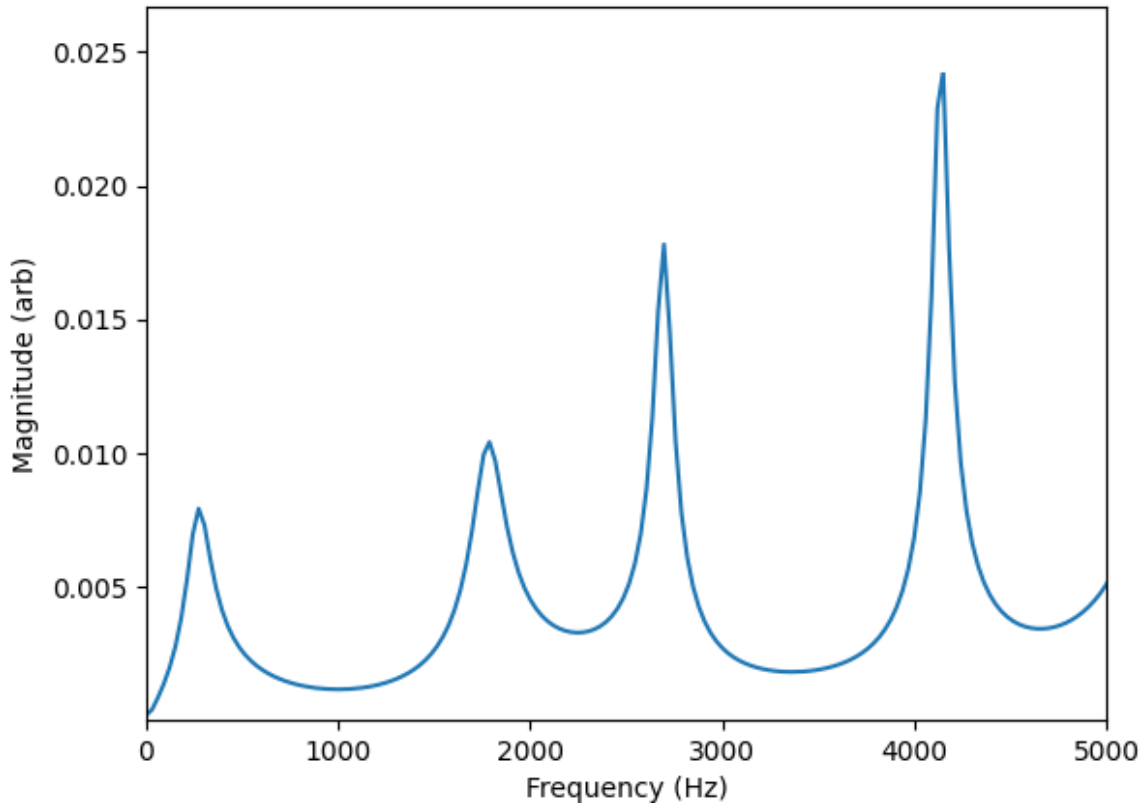


Figure 4.18: VTTF calculated from Arai /a/ model. Only the first 5 kHz are shown for clarity.

At this point there are two clear methods of producing values for formants. The first method has been described previously and involves finding the frequency of the first n peaks corresponding to the first n formants. The second method involves first producing a sound file from the VTTF by using it to filter a source signal as described in Section 4.3. This sound file can then be loaded into the Praat software package which features an algorithm for estimating formant frequencies from sound [104]. The formant calculation algorithm first resamples and then amplifies the spectral slope, which described the loss of amplitude at high frequencies, above a given frequency to further filter out high frequencies. Then the algorithm splits the dataset into a number of windows within which it applies the Linear Predictive Coding (LPC) coefficient calculation algorithm described by Burg [105] and explicitly laid out by Andersen [106].

LPC assumes that the resulting sound is produced by the filtering of an initial excitation of a buzzer with some infrequent additional sound components, and ef-

fectively attempts to solve Equation 4.15 for the roots of $a(s)$ [107]. The filter is considered to be of nth order, which corresponds to the order of the polynomial $a(s)$ from Equation 4.15 and as such is the number of poles, and has no zeros which would fully attenuate particular frequencies, meaning that the polynomial $b(s)$ has no roots. The resulting signal at each time step can be written as an auto-regression of the effect of the signal at the previous time step modulated by the coefficients related to each pole of the filter summed with the input signal at the current time step. The coefficients of these poles, or resonances, are defined by the set of linear equations that describe the output signal at each time step. If these coefficients are found for every pole, then inverse filtering the resultant signal with the calculated filter will produce the original input signal, which should either be a set of impulses at a fixed frequency, random noise, or a mixture of both. When the calculated coefficients are not accurate, the input signal will contain some of the resonance that was actually produced by the poles, and will not be spectrally flat. If the number of time steps is significantly larger than the number of poles, which is generally the case, the coefficients become significantly overdetermined. This means that there is no exact solution to the values of the LPC coefficients. Maximising the accuracy of the coefficients requires minimising the errors observed in the inverse filtered signal, usually by minimising its power through a least squares method. Once the coefficients are determined to a reasonable degree of accuracy, the roots of $a(s)$ can be calculated to obtain the frequencies at which the poles appear. The major limitation of LPC is that the system is modelled with an all-pole filter with no anti-resonances, which is likely unphysical due to the presence of side branches in the vocal tract.

The formant calculation process described above is somewhat self-defined: taking the pressure output of the simulation, using it to calculate the VTTF, producing a speech signal by using the VTTF to filter an input signal, and then running the LPC algorithm on that speech to effectively recalculate the VTTF. Still some prior research works and literature perform this same process, particularly the works by Gully [4] which provide much of the basis for this research, and it is hoped that the

error power minimisation process of the LPC will help to eliminate any issues of linear scaling in frequency described prior.

Using these two analysis methods, the data in Table 4.3 was produced. The graph of formant frequency against time produced by Praat for the recorded audio sample can be seen in Figure 4.19, and the corresponding spectrogram in Figure 4.20, as an example. Both analysis methods produce a low frequency peak, labelled 0th in the table, which lies significantly below the frequency of the first formant from the recorded audio sample. Both methods generally overestimate the formant frequencies, apart from the second formant which both underestimate. If the 0th peak in both data sets is taken as spurious, the average absolute error on a formant is 29.11 % for the VTTF peaks and 15.92 % for the Praat analysis. If the 0th peak is taken as true and the peaks from the recorded audio are shifted back to match, the average absolute error becomes 33.70 % and 29.80 % respectively. The large difference in accuracy is mainly dependent on the values for the 1st formant, which the VTTF peaks do not reproduce well.

Table 4.3: Table of possible formant values for Arai /a/ model. The first column of data, labelled 0th, shows the first peak from both data sets which may need to be excluded from the proceeding calculations. The 4th peak from the VTTF data is not visible in Figure 4.18 but was measured in the same way.

Data set	0th (Hz)	F1 (Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)
Recorded audio	-	1103	2818	3135	4565
VTTF peaks	273	1789	2697	4151	5365
Praat formant analysis	744	1241	2528	4048	5101

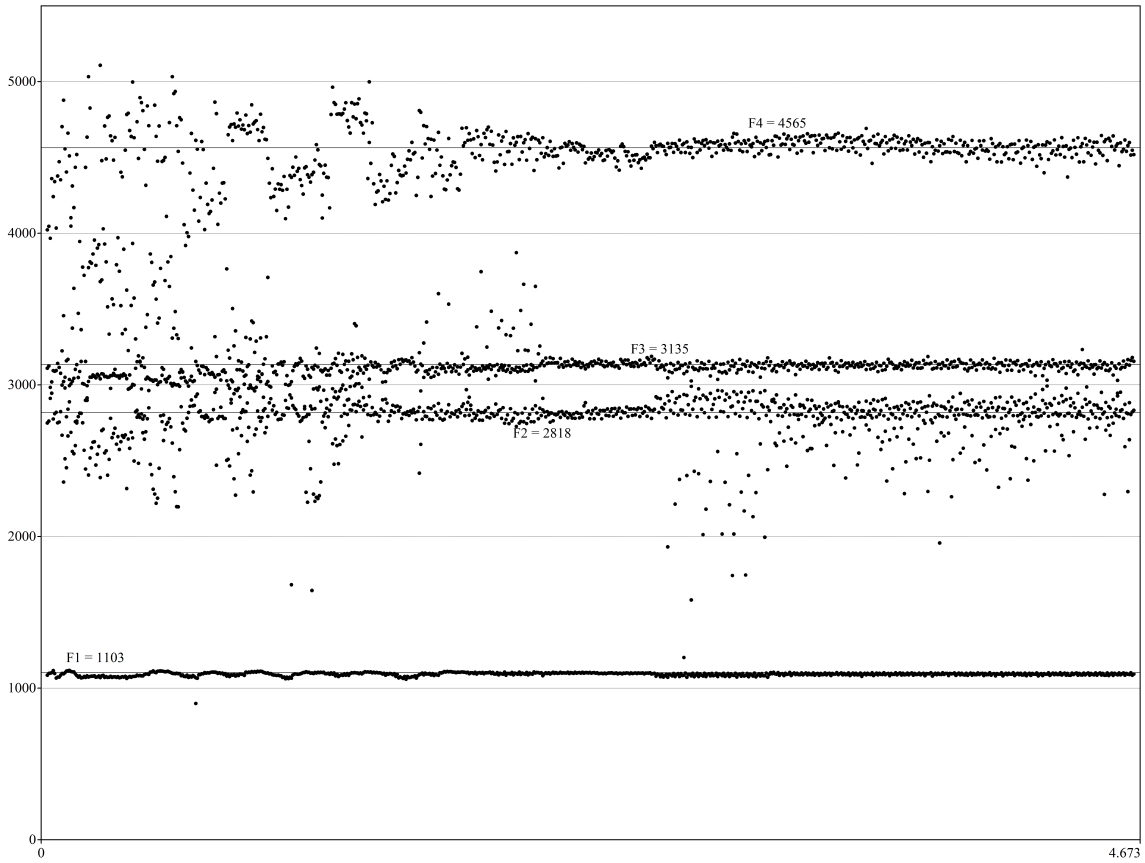


Figure 4.19: Formant frequencies in Hz against time in s produced using the Praat formant analysis algorithm on the recorded audio sample of the Arai /a/ model. Each point on the plot represents the formant frequency that the algorithm produced at a given time step. The spread of points shows that the LPC coefficients calculated in different time windows varied wildly, implying that the recorded sound either varied during the recording, or that some other sound was being picked up in the recording and combined into the LPC calculation. Four clear formants can still be picked out across the whole dataset however.

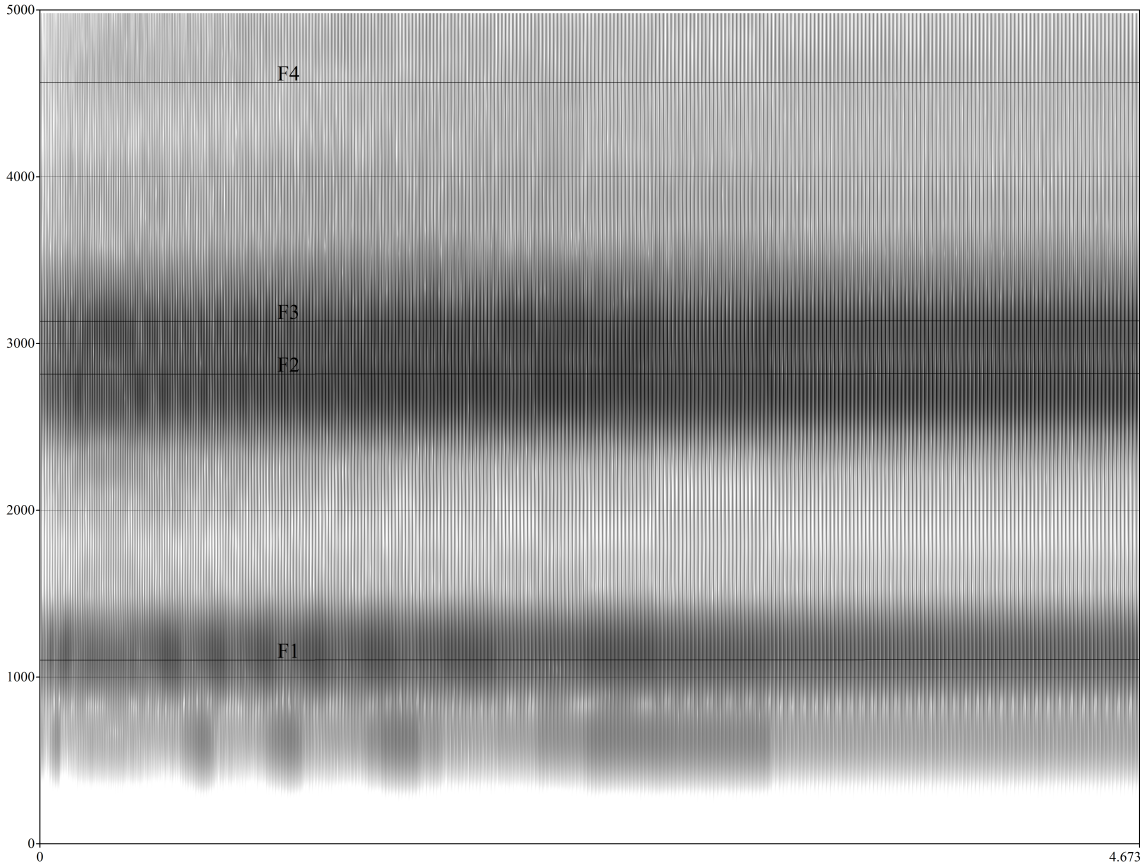


Figure 4.20: The spectrogram corresponding to the data shown in Figure 4.19. Formant frequencies in Hz extracted using the Praat formant analysis algorithm are shown as horizontal lines at their respective frequencies. Horizontal axis is time in s.

This same analysis was performed for the Arai /i/ model, as seen in Figure 4.21 and Table 4.4. Notably, the recorded values produce very high frequency formants, which is unusual and may imply some error in the recording but will be taken as truth here. Both analysis methods still produce a low frequency peak. In this model, both methods underestimate F1 and F2, then overestimate F3 and F4. Average absolute error on formant frequencies is 26.54 % and 19.07 % for VTTF peaks and Praat analysis respectively. This shows the same relationship as in the /a/ model, of the VTTF peaks producing less accurate formant values. As discussed above, the implementation of LPC here is expected to produce additional uncertainty in the output from Praat, but instead seems to improve the accuracy of the produced formant values. This may be due to an inaccurate filtering of the LF pulse with the VTTF in Python, or it may be due to the sophisticated and speech-targetted Praat

formant calculation algorithm itself. It has been continuously observed though that VTTFs produced from simulations so far are frequently laden with unexplained errors and scaling factors on the frequency, which are both avoided when only considering the Praat LPC analysis. This is still troubling, as the Praat LPC analysis is based on these VTTFs, but is likely illuminating an unknown mistake made in the production of these VTTFs that may only impact their visual presentation.

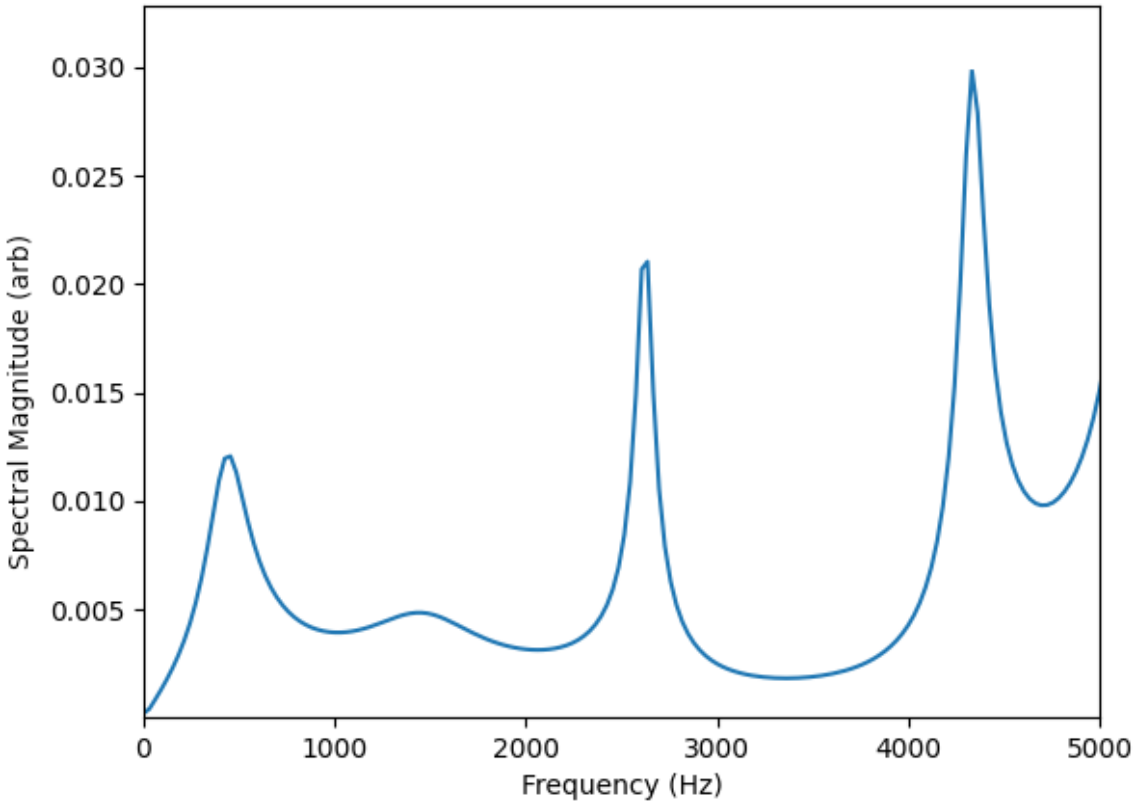


Figure 4.21: VTTF calculated from Arai /i/ model. Only the first 5 kHz are shown for clarity.

Table 4.4: Table of possible formant values for Arai /i/ model. The first column of data, labelled 0th, shows the first peak from both data sets which may need to be excluded from the proceeding calculations. The 4th peak from the VTTF data is not visible in Figure 4.21 but was measured in the same way.

Data set	0th (Hz)	F1 (Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)
Recorded audio	-	2085	2769	3142	4056
VTTF peaks	439	1439	2621	4322	5365
Praat formant analysis	917	1490	2498	3785	4765

The ease and control provided by this switch to much simpler models provides a starting point from which to attempt to improve the simulation routine by making controlled changes to the procedure and then comparing the new absolute error on calculated formants with the initial ones.

4.4 Calibration of Acoustic Simulation Algorithm Using Recorded Sound

To properly optimise the simulation routine, its outputs should be compared to physical measurements of the same tract geometry. Accuracy of this optimisation depends on the strength of the alignment between the simulation and the experimental conditions of the measurement. Using simplified linear vocal tract models, proposed by Arai as discussed in 4.3.2, should maximally align simulation outputs to real world measurements and allow for direct comparisons in accuracy while adjusting simulation parameters to maximise accuracy across a wide range of vocal tract models.

Before performing any thorough studies on the variation of parameters, there are a few potential changes the simulation procedure itself which can be explored. The first potential change is in accounting for the effect of reflections from the domain boundaries on the pressure within the simulation. Figure 4.22 shows pressure data in the empty space outside the lips at intervals of 0.2 cm for the first 3.3 ms after the initial volume velocity input. The scaling of the pressure amplitude has been allowed to vary for each plot and the data has also been trimmed at the start to sync up each data set to the point in time that it first received pressure from the initial pulse. In homogenous space that is at a sufficient distance from any other reflective features like the lips, the only effect that should be visible on a pressure wave in this simulation over distance should be power loss due to the inverse square law. This would appear as a scaling factor applied to the entire pressure node. While the earlier nodes may be too close to the lips to properly obey this, this behaviour

should become more visible the further from the lips that the samples are taken. It can be clearly seen that the shape of the pressure trace does change at each node, which implies that reflections from the domain boundaries are having a measurable effect on the simulation output. The effect of this on the output of formants has not yet been explored, but it is expected to result in a noticeable difference in VTTF produced at successive nodes.

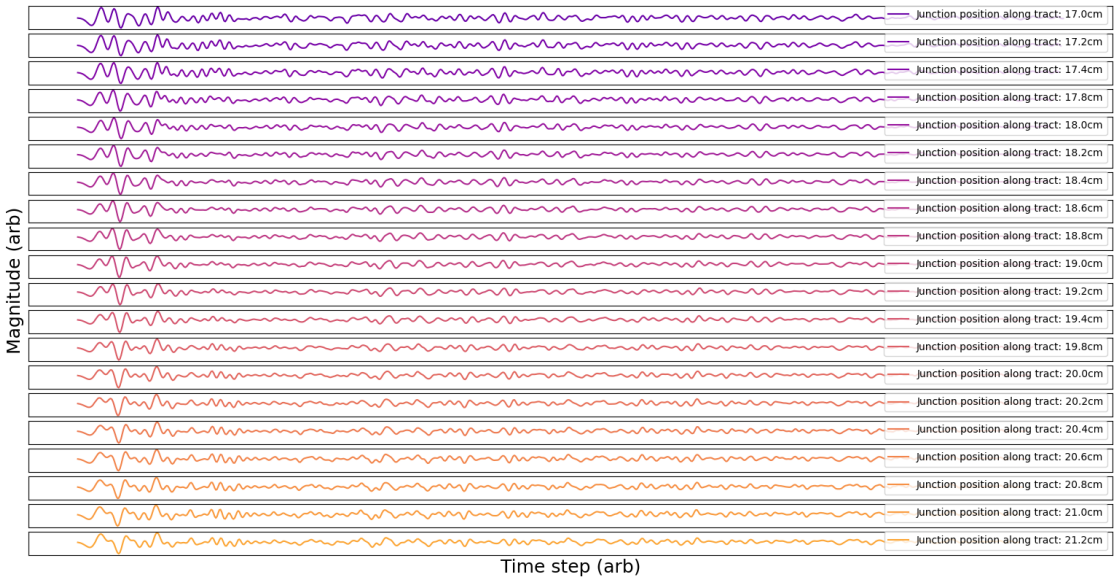


Figure 4.22: Pressure data for the first 3.3 ms for nodes in empty space outside the lips, at intervals of 0.2 cm. The start of each data set has been trimmed to approximately sync each data set to the point in time that the initial pressure pulse reached the node. The scaling of the pressure amplitude has been allowed to vary on each plot to better illustrate the shape of each plot. Nodes towards the bottom of the plot are further in to empty space.

During research into other methods of accounting for pressure loss at the boundaries, a possible error was discovered in the DWM update equation that was currently in use for propagation within the model. The FDTD update equation for free space contains a term proportional to the pressure at the current node at the current time step: $2(1-2\lambda^2)p^n_{l,m,n}$ where λ is the Courant number. There is no similar term in the DWM formulation described in previous literature and so it was proposed that, as DWM and FDTD are technically equivalent methods, a term dependent on the current pressure could be needed. By inspection, this term was added in to the DWM update equation with a suitable scaling factor as $+2(1-2\left[\sqrt{3}/3(\sqrt{3}+G_{sum})\right])p^n_{l,m,n}$.

Simulations including this missing term exhibited rapid solution growth which destroyed the ability to resolve the 'real' variations in pressure caused by the geometry of the model. In the DWM formulation of the simulation domain the scattering junctions themselves are not necessarily considered as containing an amount of acoustic pressure, and instead they act as an interaction point between the connected waveguides that hold an instantaneous pressure used for calculating propagation into the waveguides for the next time step. As such, including the pressure in the scattering junction in the calculation of the pressure in the junction for the next time step effectively leads to a double counting of the pressure which behaves like a DC pressure offset at every scattering junction similar to that seen in soft-source solution growth. The inclusion of this term in the pressure update equation is as such incorrect and non-physical, and it will not be included in any future simulation algorithm.

4.4.1 Variation of Physical Parameters

The simulation algorithm in its current implementation has only a few free parameters that can be varied before runtime. The most important of these free 'parameters' is the update equation itself. The Update Equations 2.15, 2.16, and 2.17, in combination with the Domain Boundary Update Equation 4.11, are deemed here as accurate based on their derivations. Within the update equations are a variety of physical parameters including acoustic admittances between the walls of the tract and the air within the tract, and the speed of sound in air which is approximated to be the same as the speed of sound within the walls of the vocal tract. These values are taken from other literature and are also considered as appropriate despite any in-built assumptions and inaccuracies to the real system. Specifically, the speed of sound through the vocal tract walls themselves would be significantly higher than in air, meaning that propagation through the solid parts of the model is inaccurately slow.

The first parameter that is considered valid to be varied is the resolution of

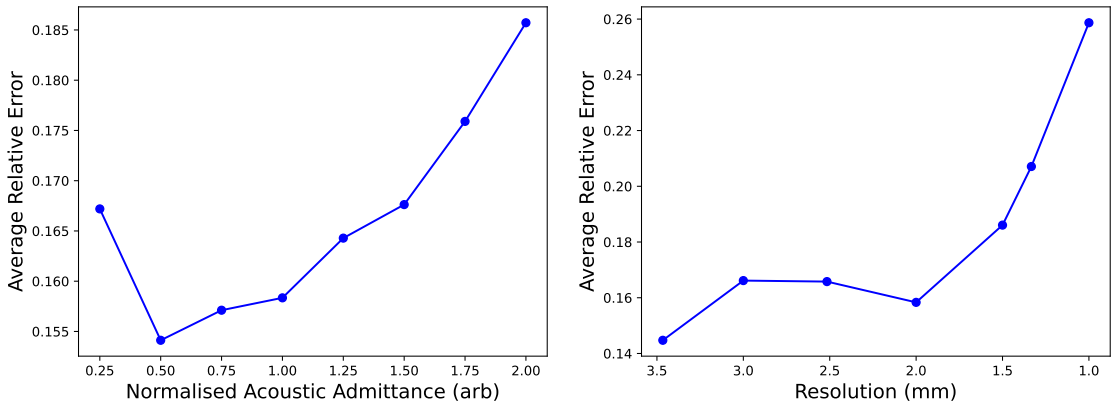
the geometry of the vocal tract. The geometry of the model has a mathematical impact on the simulation routine through the Courant convergence condition. The Courant number is given as $\lambda = c_{air}\Delta T/\Delta X$, in the form of the spatial step ΔX and the temporal step ΔT. Numerical stability is maximised when the Courant number is fixed at $1/\sqrt{D}$ where D is the dimensionality of the system, and so variation of the spatial step sets the temporal step ΔT of the simulation. The temporal step sets the sampling frequency of the simulation, with small temporal steps requiring very high sampling frequencies. With a spatial step of 2 mm, the temporal step is set at 3.36 µs giving a sampling frequency of 297 912 Hz. This is extremely high, but lowering this frequency while keeping the same model resolution will violate the Courant convergence condition and introduce numerical instabilities in the simulation routine. Increasing the resolution of imported geometry would allow for the sampling frequency to be greatly reduced: Moving from 2 mm to 3 mm resolution would reduce the sampling frequency required by approximately 100 kHz but would also reduce the accuracy of the imported geometry.

The second major variable parameter is the normalised acoustic admittance between the simulation domain and the far-field, found in the equation for the boundary updates. This parameter models the amount by which the flow of acoustic pressure out of the simulation is impeded. Ideally, there would be no impedance at the boundary and any pressure incident on the boundary would flow out of the model without any reflection, corresponding to a normalised admittance of 1. With this implementation however, it is impossible to reach this ideal situation as whenever there is a discrete variation in acoustic admittance there will always be an amount of scattering of that boundary. Improved formulations of discrete wave-based algorithms here have successfully managed to minimise this scattering by making use of a Perfectly Matched Layer (PML) around the edge of the domain [108, 109]. To accomplish this, the space co-ordinates within the PML undergo a transformation into complex co-ordinates. Under this transformation, propagating waves become exponentially decaying ones meaning that with a sufficiently long PML, the exact

form of the pressure variation remains continuous and drops off to zero without any discontinuity scattering.

Unfortunately this PML formulation is dependent upon the behaviour of multivariate vector fields, like the E and B field in Maxwell's equations, under those transformations. The DWM method used here only concerns the pressure field in the domain and so the PML is not easily implemented here. As such, with the current set of update equations, scattering off of the boundaries of the simulation domain is inevitable. It may be that this scattering could present opportunities to improve the accuracy of results when it is considered that they will be applied to real environments. By carefully choosing a value for the normalised admittance at the boundaries to the simulation domain, some phenomena or potential reflections that would otherwise be ignored in the simulation could be accounted for. An example of this may be found in modelling reflections off parts of the head which are outside the boundary as reductions in the boundary admittance.

The majority of work improving the simulation in this section has concerned producing a large breadth of simulation data on the various Arai vocal tract models with varying resolutions and normalised acoustic admittance values. In the figures referred to in this section, the data points represent the average difference between each calculated formant value and its corresponding real world recorded value for the first five formants.

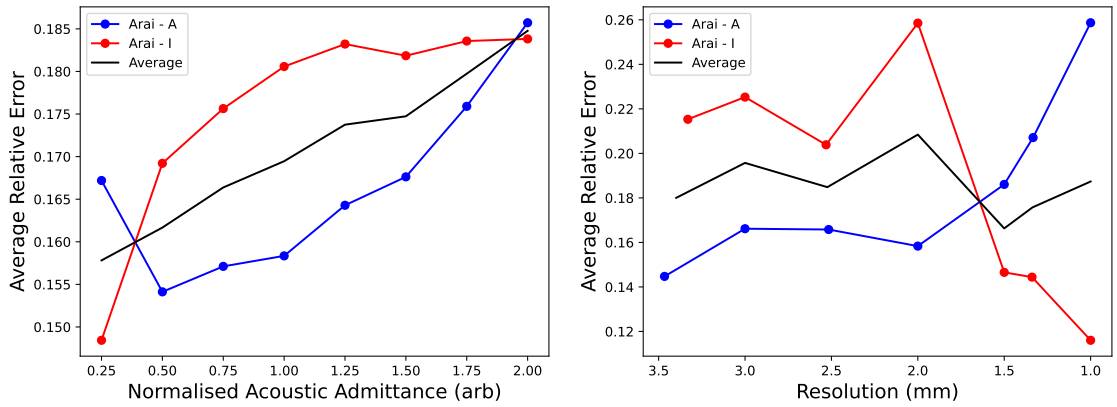


(a) Varying normalised admittance.

(b) Varying resolution, in mm.

Figure 4.23: Studies of Arai /a/ vowel model simulated while varying normalised acoustic admittance and resolution.

Figure 4.23a shows the effects of sweeping the normalised acoustic admittance G from 0.25 to 2. We observe a minimum absolute error in formant values at G=0.5 here with any value from 0.5 to 1 resulting in a relatively low contribution to the error. Values outside this range cause increasing error the further from this range. Another set of simulations were performed whilst varying the input model resolution, shown in Figure 4.23b. This plot shows the highest accuracy in the produced format values at the lowest resolution, with error increasing with resolution apart from at 2 mm which shows a slight dip in average percentage error. This behaviour is surprising, as it was expected that low resolutions would lead to a misrepresentation of the model geometry and that sufficiently high resolutions past the actual resolution of the scans the model was obtained from would also lead to the appearance of spurious additional scattering surfaces. As these studies were not considered fully conclusive, the same process was undergone again on the Arai /i/ model, with Figure 4.24 showing the combined results of the two models for comparison.



(a) Varying normalised admittance.

(b) Varying resolution, in mm.

Figure 4.24: Studies of Arai /a/ and /i/ vowel models simulated while varying normalised acoustic admittance and resolution. The average error across both models is shown in black.

Figure 4.24a shows the effect of varying the normalised acoustic admittance. Both models follow the same overall trend of increasing normalised admittance leading to a greater error on formants, apart from at 0.25 G in the Arai /a/ model. While this value may be spurious, the value at which the two data sets cross, G = 0.389, was chosen as a compromise between the two models. Figure 4.24b continues to show unexpected relationships between the resolution of the input model and the formant accuracy. While the first model showed a general increase in accuracy as resolution was decreased, the Arai /i/ model shows decreasing accuracy as resolution is decreased. Assuming that this variance in behaviour is not due to an inaccuracy of the simulation routine, this implies that formant accuracy may be very dependent on the geometry of the input model itself and how that geometry falls on the grid used to perform the voxelisation. The average of both datasets was minimised at 1.5 mm, but more investigation was deemed to be necessary before reaching a valid conclusion.

Before applying these values for the normalised admittance and model resolution, it is also valuable to consider the contribution to the overall error of varying these two parameters. Across both models, the maximum range of error on formant values seen when varying the normalised admittance is approximately 4 % and the value chosen

for future simulations has an average percentage error across both models of 16 %. Resolution on the other hand showed a maximum range in errors of 14 % with the value selected having an average percentage error of 17 %. This much larger range of error when varying resolution implies again that accuracy is largely dominated by geometry in these simulations, and that setting an appropriate resolution is much more impactful than the value of the normalised acoustic admittance.

Notably, both of these plots have been produced while considering the lowest frequency formant outputted by LPC formant analysis to be a 0th 'false formant', first mentioned in Section 4.3.2, which may be caused by the process of creating the sound files that the formant analysis uses. When this formant is included and the previous F5 is ignored, average percentage error increases by 12 %. While the results obtained when including the false formant are much less accurate, they do produce relationships between the free parameters and formant accuracy which are much more intuitively shaped. For varying G values, a continuously increasing amount of error in formant values with increasing G is observed, with no outliers. For varying resolution, an overall downward trend on formant errors with increasing model resolution is produced, with both simulations reaching their highest accuracy at the highest simulated resolution of 1 mm. At this point, as the accuracy of all outputs that ignore the false formant is higher than almost any of the data points produced when including the false formant, the relationships shown in the false formant included data will not be considered.

Having produced a suspected ideal value for the normalised acoustic admittance and model resolution parameters using just two of the Arai models, these parameters were applied to the remaining three models to test their efficacy without a worry of over-fitting to the test cases. The three remaining models were simulated with resolutions of 1.5 mm and 2 mm, and with G values of 1 and 0.389. No intermediate values were simulated at this time. For all three of the remaining models the highest accuracy in produced formants is seen with a resolution of 1.5 mm and a normalised admittance of 0.389. This directly agrees with the findings of the studies on the

Arai /a/ and /i/ models.

The final 'free parameter' to be explored within these simulations is that of simulation duration. Simulations need to be long enough to include enough acoustic propagation data to model the full behaviour of the tract. After a certain point in time however, most of the remaining acoustic pressure variation is dominated by non-physical pressure contributions such as that of reflections from the domain boundaries and solution growth. Also, simulations over a long time period take an increasingly large amount of memory, storage, and computation time. Minimising the required simulation time to produce an accurate output is important both for simulation accuracy and for usability.

A series of formant values were produced using a varying amount of simulation data, ranging from the first 1 ms up to 20 ms. Both models showed the lowest error on formant values with a simulation duration of 5 ms. This is six times fewer time steps than most of the simulations performed throughout this work so far. This time duration also seems physically appropriate. The length of the Arai vocal tracts range between 0.156 m to 0.168 m based on measurement of the models. Unimpeded sound would take approximately 0.45 ms to 0.49 ms to travel from the glottis to the lips. As such, 5 ms would account for the first five full reflections of the acoustic pressure between the glottis and the lips, which should include most if not all the relevant acoustic propagation effects. With an initial impulse pressure of 1 in arbitrary units of pressure, the maximum acoustic pressure in the simulation domain drops by two orders of magnitude after 3 ms. Finally, the length of the LF pulse which is commonly used for source-filter methods of speech synthesis is on the order of 10 ms meaning that any pressure data produced after that amount of time in the simulation will not account for any interaction with the new incoming pressure field. For real vocal tract models, the vocal tract's mid-sagittal axial length is in the range 0.235 m to 0.3 m and unimpeded sound would take 0.68 ms to 0.87 ms to travel from the glottis to the lips. As such a simulation length of 5 ms in these models would contain roughly 3 full reflections between the glottis and the lips.

4.4.2 Reanalysis of Arai Data with Corrected VTTF Calculation

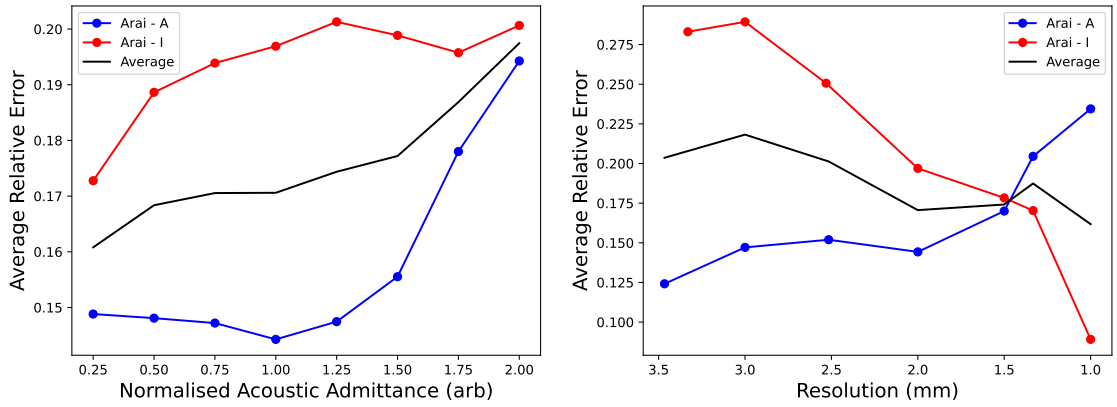
During this study of simulation duration, an error in the analysis method was discovered. The formula for the volume velocity source which is played from the source node during the simulation is given as

$$u(n)_{in} = e^{-((\Delta T \cdot n - T_0)/0.29T_0)^2}, \tag{4.16}$$

where n is the index of the current time step ranging from 0 to the number of steps required to reach the desired simulation length, ΔT is the time step of the simulation, and $T_0 = 0.646/f_0$ with f_0 as the desired frequency range to produce suitable excitations across, here set to 20 kHz. The labelling in this equation is different to that given in the source material it is pulled from, which was mimicked to provide more points of comparison [110]. In the interpretation of the original labelling, an additional factor of T_0 was mistakenly inserted into the exponent during analysis which was not present during the simulation. This led to VTTF outputs during the prior studies of optimal simulation parameters discussed in Section 4.4.1 that did not represent the simulation accurately.

Reanalysis of the data created using the linear Arai vocal tracts produced similar results without some of the unexplained features. The corrected graphs of average formant accuracy for varying G and model resolutions can be found in Figure 4.25. There is no longer an overlap between the two data sets within the range of G values explored in this study and the minima in formant error in the Arai /a/ data set is significantly reduced in magnitude compared to its neighbouring points and shifted in G. This new analysis shows that both models benefit from a low value of G, with the Arai /a/ data set having a relatively constant formant error below G = 1.25. For the variation of model resolution, the strange behaviour at a resolution of 2 mm has now vanished in the Arai /i/ data set, giving a more linear relationship in the data set between increased resolution and decreased error. Unfortunately the two

models still show opposing relationships with a crossover point at 1.5 mm. We also observe significantly larger ranges of absolute error percentage with the maximum range of percentage error across normalised acoustic admittance values at 5 % and the maximum range of percentage error across model resolutions at 20 %. Minimum average percentage error does still occur at around 16.5 %.



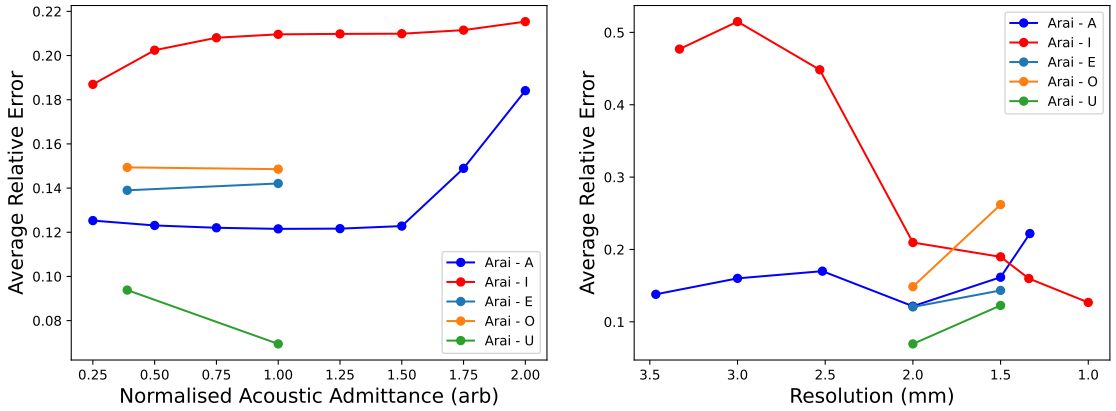
(a) Varying normalised admittance.

(b) Varying resolution, in mm.

Figure 4.25: Reanalysis of studies of Arai /a/ and /i/ vowel models simulated while varying normalised acoustic admittance and resolution. The average error across both models is shown in black.

A final analysis of all five of the Arai data sets was done, this time only concerning the first 5 ms of pressure data as discussed earlier in this section, and can be found in Figure 4.26. The aim of this analysis was to produce finalised optimal parameters for acoustic simulation of real vocal tract models based on these simplified linear models. For variation in normalised acoustic admittance a slight reduction in the maximum range of percentage error shown across one model is seen, and it is also observed that the percentage error distribution is a lot smoother across a large range of G values. Figure 4.26b shows a minimal change in the analysis for the /a/ data, but a large error is introduced in the /i/ dataset above a resolution of 2 mm. While this spike in percentage error is currently unexplained, the conclusion to be drawn from these new analyses is that the long-time effects that were assumed to be affecting simulation output were present as expected. Considering all five models, for variation of both normalised admittance and model resolution, the lowest average

percentage error is achieved with an admittance of anywhere in the range 0.25 to 1, with 1 being chosen for future research due to its physical underpinning, and model resolution in the range 1.5 mm to 2 mm with 2 mm being chosen due to resolution of the input scans.



(a) Varying normalised admittance. (b) Varying resolution, in mm.

Figure 4.26: Reanalysis of studies of Arai /a/, /i/, /e/, /o/, and /ɯ/ vowel models simulated while varying normalised acoustic admittance and resolution.

4.4.3 Optimised Simulation Parameters for Real Vocal Tracts

Acoustic simulations using the simulation routine previously described were performed on two human vocal tract models produced using MRI scans of the subjects. One model is that of a male speaker producing the /ɜː/ phoneme found in the word 'Stern', and the second model is that of the preserved vocal tract of the mummified remains of Nesyamun, 'True of Voice'. As discussed in Section 4.3.1, the 'Stern' model was selected due to its availability and as a representation of an intact living human tract, and the Nesyamun model was picked due to its availability and due to interest based on previous works by this group [8]. For the first round of simulations the voxel size used to prepare the model for the simulation was set to 2 mm, the normalised acoustic admittance at the simulation domain boundary was set to 1, and the simulation length was set to 5 ms.

The average percentage error on the four formants extracted from the simulation of the 'Stern' model was 29.6 %. The error from the five formants extracted for the

Nesyamun model was 23.9 %. These values are too large to ensure that the output they represent is true to the real versions of these tracts. Also, a large majority of this average error comes from a drastic overestimation of the first formant, with the simulation output often showing double that of the recorded sound. It should be noted that the recorded sounds being used to compare the simulation to the original tracts is produced by fitting 3D printed recreations of those tracts onto speakers which then play a glottal Liljencrants-Fant (LF) source, which is also used as the target of the Vocal Tract Transfer Function (VTTF) filter produced by the simulation. This tool, the Vocal Tract Organ, was developed by Howard [111] and is made thorough use of in this research. As such there may be parameter differences between the simulation and the prints, like wall thickness and materials, that affect the output. It is believed at this point that these differences should have a relatively small effect on sound production.

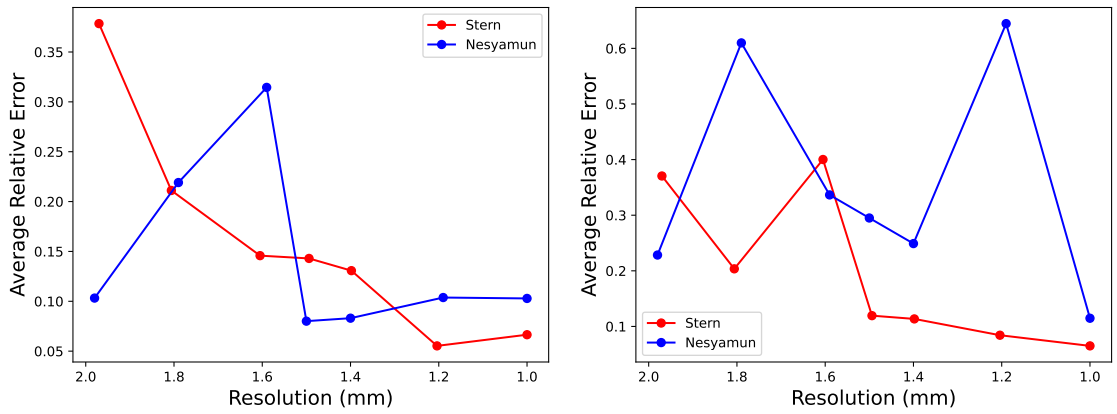
One of the original driving factors behind the use of linear vocal tract models for simulation optimisation was in reducing the time needed to perform the simulation, possibly by reducing resolution or simulation duration. As it has been determined that drastically shorter simulation times still produce equivalent, if not better, formant accuracy, running a battery of simulations to test parameters on real vocal tract models is now deemed to be more viable. One benefit of using a high level programming language like Python is that it strictly keeps the memory allocation for different processes separate unless explicitly requested, which means that multiple instances of the simulation can be run at the same time on different models with different parameters given enough computational power. Simulation speed was observed to only be slowed by approximately 20 % when running up to four simulations at once, with that slow down improving as the simulation routine received additional development.

Initial explorations into several variations were investigated during these new simulations, with more thorough studies presented in Subsection 4.4.4 below. Varying resolution between 1.5 mm to 2 mm showed significantly improved accuracy at

higher resolutions with 1.5 mm resolutions gaining 13 % to 20 % accuracy over 2 mm ones. Decreasing normalised acoustic admittance also led to gains in accuracy, with the lowest tested value of 0.389 producing the highest accuracies. Finally, it was noticed that additional space for propagation of the acoustic pressure after it exits the lips was being added to the wrong axis, meaning that no space was being added in the correct direction. This was corrected and then experimentation was done with moving the point at which the formants were measured further away from the lips. Both of these changes improved the accuracy of formant measurements. The conclusions above represent approximately 25 full simulations, during which small changes and improvements were made to the simulation routine itself. At the conclusion of these simulations, the average percentage error for the 'Stern' model was minimised at 12.18 % with a resolution of 1.5 mm and a normalised acoustic admittance of 0.389. For the Nesyamun model, with a resolution of 1.5 mm and a normalised acoustic admittance of 0.3, the error was minimised to 6.64 %. This same average error parameter in previous similar work by Gully [4] was given as 14.01 % and deemed to be equivalent to if not an improvement on several existing 3D acoustic vocal tract simulations. The work by Gulley is more rigorous than this work at this point, only having concerned two vocal tracts of which one is thoroughly dessicated and does not strongly resemble the internal geometry of a living human vocal tract, but the higher accuracy seen here is still very promising.

4.4.4 Quantitative Study of G and Model Resolution Variation

This subsection contains a thorough quantitative analysis of the data that led to the conclusions presented in Section 4.4.3 above. Each data point in Figures 4.27 and 4.28 represents a 5 ms acoustic simulation with a different set of parameters.



(a) Varying resolution at G=0.35. (b) Varying resolution at G=0.65.

Figure 4.27: Average relative error in formant values at varying resolutions in the 'Stern' and Nesyamun vocal tract models, at two values of the normalised acoustic admittance.

Figure 4.27 shows the effect of granular changes in resolution on the output formant accuracy. This effect was explored at two different values of the normalised acoustic admittance as it was unclear at the time of simulation as to which value may be more appropriate for simulation accuracy. The data in both cases shows vastly different distributions, both between models at the same normalised admittance value and the same model at different normalised admittance values. For the former, it had already been proposed that simulation accuracy was highly dependent on model geometry and these data sets seems to corroborate that conclusion, with a change in resolution of 10 % producing a 50 % loss in average accuracy in the most extreme cases. For varying of normalised admittance while holding resolution constant however the approximate range of average error remains similar but the distribution varies. Ignoring the real physical representation of the normalised acoustic admittance at the domain boundary, from the perspective of the simulation algorithm it may be the case that the optimised value of the normalised admittance varies at different resolutions. This may imply the existence of some kind of combined parameter that relates to the properties of the physical space the simulation is occurring within. As such, the discontinuous peaks seen in some of these graphs may actually be a consequence of only plotting a cross-section of the 3D surface of

normalised admittance, resolution, and average percentage error.

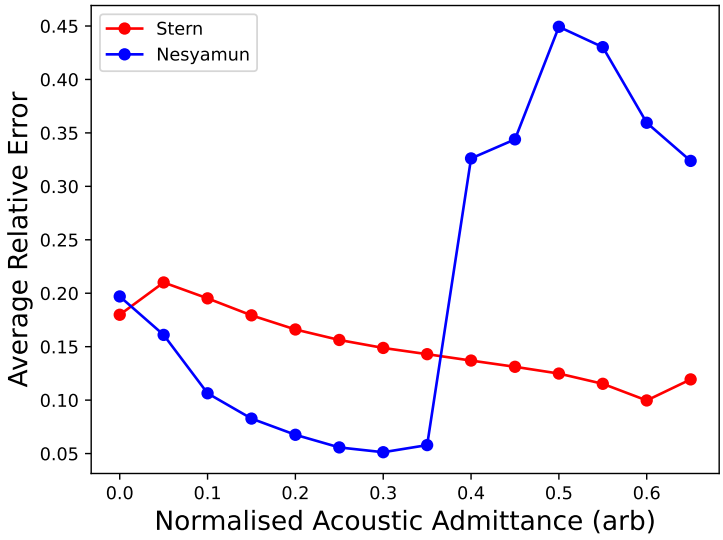


Figure 4.28: Average relative error in formant values at varying normalised acoustic admittances. Model resolution is set to 1.5 mm.

Figure 4.28 shows the effect of the normalised acoustic admittance on the average percentage error of extracted formants for both vocal tract models at a model resolution of 1.5 mm. The two models show two very different behaviours. For the 'Stern' model, a general decrease in error on formants is seen as the normalised admittance is increased. At G = 0.65 there is a slight up-tick in error, but this may not be characteristic of the behaviour if this graph was extended in G. For the Nesyamun model the magnitude of the error initially decreases steadily with increasing normalised admittance and then jumps upwards to almost 50 % before beginning to steadily decline again. No obvious explanation, apart from the theory suggested above about the combined effect of resolution and normalised admittance, presents itself to explain this behaviour.

The lack of a strong conclusion from the simulations run thus far requires some consideration at this point. Frequently, the ideal parameters from the 'Stern' model have not fully agreed with the parameters that provide the best results for the Nesyamun model. With the apparent effect of geometry on the accuracy of the simulation, it is important to consider the key physical differences between these two tracts. Figure 4.1 shows a side by side image of the two models. The 'Stern'

model is of a living speaker lying down inside an MRI machine whilst producing the corresponding vowel sound. The Nesyamun model was produced using scans of the vocal tract of the preserved body of the 3000-year-old Egyptian mummy. The Nesyamun model is fully dessicated, has its tongue protruding from its mouth, and is not in any particular vowel articulation. As can be seen in the images of the tract, the Nesyamun model also has fully resolved teeth, whereas the teeth of the 'Stern' model are completely missing. Finally, the Nesyamun model likely also has further muscle decay in addition to the lack of a tongue. These differences amount to the air passage of the Nesyamun model being very open, in comparison to the 'Stern' model which is quite constricted. Some living vocal tract models are in fact so constricted that the air passage can become fully occluded during the voxelisation procedure. This could be prevented by increasing resolution of the voxelisation, but the required resolution would be at least twice as small as the smallest constriction in the model to guarantee that the constrictions are properly resolved. Determining the size of the thinnest constriction is itself a complex problem. The use of DWM as the algorithm for this simulation is a benefit here in that the implementation of the acoustic admittance within the domain allows acoustic propagation even within the walls of the model, which are not modelled as lossy materials. Some scattering will arise at these occlusions however, as described in Section 2.1, this is physically accurate.

In this research so far, the Nesyamun model has tended to produce less error in its formant values when compared to the recorded sound produce by the 3D printed model which has led to parameters that work well for it being favoured over parameters that work better for the 'Stern' or Arai models. This preference for the Nesyamun model may have led to an over-fitting to its particular geometry that may make the simulation routine less appropriate for its intended goal of modelling the voice output of living speakers. At this point it was decided that significantly less weight would be placed on the outputs of the Nesyamun model and that a new set of models would be acquired for further testing alongside the 'Stern' model.

Vocal tracts producing the English phonemes for /iː/ as in ‘Neap’, /uː/ as in ‘Food’, and /ɑː/ as in ‘Hard’ have been acquired and implemented into the simulation routine. Some tests were undertaken on the produced formant accuracy for these models and the ‘Stern’ model at 1.5 mm, but it was quickly decided that, as in the data from Figure 4.27, 1 mm provided better accuracy. It is possible that even higher resolutions would give even higher accuracy, but the sampling frequency required to fully resolve a domain with a 1 mm resolution is already almost 600 kHz and further increasing resolution would lead to ballooning simulation times and the potential introduction of further high frequency errors.

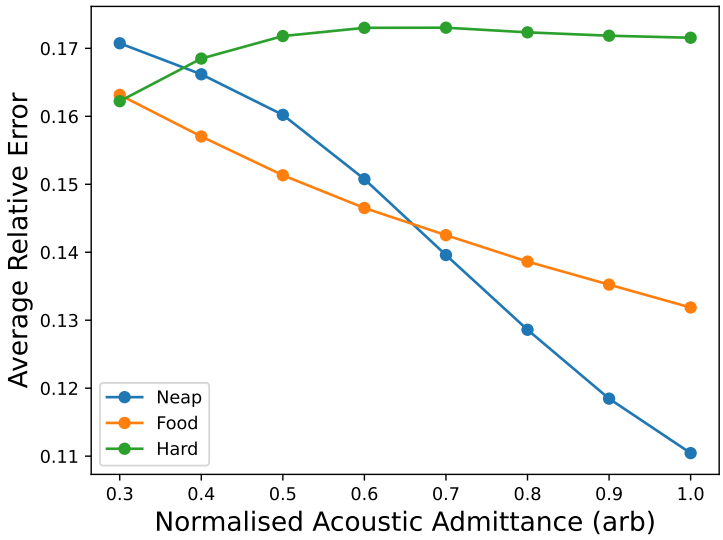


Figure 4.29: Average percentage error in formant values at varying normalised acoustic admittances for the ‘Neap’, ‘Food’, and ‘Hard’ vowel vocal tract models. Model resolution is set to 1 mm.

Figure 4.29 shows the effect on the average percentage error of produced formants of the ‘Neap’, ‘Food’, and ‘Hard’ models of varying normalised acoustic admittance. All three models show no discontinuities, like those seen in the Nesyamun model, as expected. The ‘Food’ and ‘Hard’ models show increasing accuracy with increasing normalised admittance, while the ‘Hard’ model has a slightly higher accuracy at low values of the normalised acoustic admittance but with a relatively flat distribution across the whole range. The different behaviour observed with changing normalised admittance between the Arai models and these more realistic vocal tract

models is likely due to the different shapes of their simulation domains. The Arai /a/ model has an aspect ratio of 50:50:156 and the 'Neap' model has an aspect ratio of 60:103:115. With a source placed at low x, y, and z co-ordinates, the Arai model will have many reflections off the domain boundary in both x and y present in the output, whereas the 'Neap' model will only have a similar number of reflection in the x-axis. As the normalised admittance is the sole parameter controlling the behaviour of reflections at the boundaries, the nature and frequency of those reflections would likely have a large effect on the ideal value of the normalised admittance. These aspect ratios are approximately the same for each model of the respective types. Another potential factor affecting this relationship is in the materials that the physical models being used to validate these results are made from. The Arai models are constructed by milling out regions of an appropriate diameter based on measurements of the vocal tract from acrylic cylinders with a 50 mm diameter. The maximum diameter of the milled out sections is stated as 38 mm [11]. As such, the minimum wall thickness is 6 mm radially. The physical vocal tract models are 3D printed from Polylactic Acid (PLA) and are given a thickness algorithmically within the modelling program in use of approximately 2 mm. This added wall thickness, in addition to the likely different acoustic impedance of these two materials, results in different amounts of 'through-wall' transmission in these models, possibly requiring different normalised admittance values to accurately model them.

For the remainder of this research, and indeed so far, finalised measures of accuracy will generally be given in the form of the Mean Absolute Error (%) (MAE), which is obtained by first finding the percentage error of each produced formant when compared to its recorded counterpart, and then taking the average of the absolute values of those percentage errors. When discussing the accuracy of the simulation across multiple models however, there is more than one method of calculating a final value. The two methods considered here are to take the average of the MAEs of each tract model's outputs, and to instead take the MAE of each formant across all the models in question and then take the average of those MAEs.

The first method is closer to a representation of the average accuracy of the whole simulation output whereas the latter is more closely tied to the average accuracy of the production of each formant. To remain consistent with the method used to produce values given in other research, the latter method will be used.

At this point it was deemed that a resolution of 1 mm and a normalised acoustic admittance, G, of 1 provided good accuracy across all 4 models. The average of the MAE of each formant produced in simulations of the 'Neap', 'Food', and 'Hard' models is 12.77 %. It is proposed that, as no special care was given to the geometry of these models in their simulations, this percentage error is similar to that which one could expect for any vocal tract model simulation. This value is still similar to, or less than, the values quoted by Gully [4, 112] and is low enough to corroborate the validity of outputs from this simulation routine.

Finally, a brief investigation was undertaken on the values used for acoustic impedance within the simulation routine. As discussed in Section 4.3, Wang et al. [97] and Arnela et al. [24, 95] provide different values for the acoustic impedance of air in the vocal tract and the walls of the vocal tract itself. The former uses an air density of $1.17\,\mathrm{kg\,m^{-3}}$ and a speed of sound in air of $346.3\,\mathrm{m\,s^{-1}}$ which gives an acoustic impedance of air, Z_{air}, of $405.171\,\mathrm{kg\,m^{-2}\,s^{-1}}$ per unit volume, which is more commonly given with equivalent units of $405.171\,\mathrm{Pa\,s\,m^{-1}}$. Wang et al. also uses a vocal tract wall density of $1000\,\mathrm{kg\,m^{-3}}$ and a speed of sound within the vocal tract wall of $1500\,\mathrm{m\,s^{-1}}$ which gives an acoustic impedance of the vocal tract wall, Z_{wall}, of $1.5\,\mathrm{MPa\,s\,m^{-1}}$. These values were obtained from measurements and likely include bone conduction in the values pertaining to the walls. Arnela et al. provides an air density of $1.14\,\mathrm{kg\,m^{-3}}$ and an air speed of $350\,\mathrm{m\,s^{-1}}$ which produces a Z_{air} of $399\,\mathrm{Pa\,s\,m^{-1}}$, and states a Z_{wall} of $83\,666\,\mathrm{Pa\,s\,m^{-1}}$ from prior research [96]. This low value for the wall impedance is surprising as they imply that the speed of sound in the wall of the vocal tract, or the density of human tissue, is not greater than in air by more than two orders of magnitude. Both sets of values, when used in simulation, provide very similar accuracies. Using the Wang et al. values produces

an accuracy loss of 0.22 % compared to the Arnela et al. values. This difference is much smaller than the magnitude of difference in accuracy produced by the other changes discussed in this work and so is deemed to be not as important. The Arnela et al. values will be used to minimise the size of any discontinuities in the domain to alleviate any numerical errors.

4.5 Chapter Summary

This chapter presents the large body of work done during this research in the creation and development of an acoustic propagation algorithm in Python. Python was chosen as the language of choice for this research due to its ease of use, its open source nature, and its wide library of community modules.

Preliminary exploration into the methods of importing 3D geometry into a simulation domain led to the use of voxelisation not only for geometry imports, but also as a visualisation tool. Voxelisation made the field of wave-based algorithms which step across a Cartesian unit grid accessible, and so a Finite-Difference Time-Domain (FDTD) method was explored for the acoustic simulation in this work. A series of errors in the implementation and shortfalls of the method led to its abandonment, replaced by a Digital Waveguide Mesh (DWM) method. This method solves the wave equation directly while providing a powerful simplification in the definition of the model geometry, allowing for acoustic simulation throughout the entire domain without needing to apply special boundary updating procedures at the walls of the vocal tract.

Once completed, the acoustic propagation simulation routine was validated against a number of 3D printed physical vocal tract models extracted from Magnetic Resonance Imaging (MRI) data of living subjects. The simulation routine showed some agreement in its outputs with audio recordings from the physical tract models, but with some unresolved scaling factors and other such errors. It is currently believed that the source of this scaling errors is either in the failure of the Fast Fourier Transform (FFT) procedure due to a breakdown of linearity in the model, in a lack of

normalisation of the outputs, or in a lack of sophistication in the analysis routine.

Further investigation of these errors is only possible with a simulation routine which is strongly aligned to the physical domain it is attempting to simulate. To achieve this condition, a set of five linearised vocal tract models were obtained and used as the basis for a round of optimisation studies of this acoustic simulation. These studies considered a variety of implementation changes and parameter variation, notably in the normalised acoustic admittance at the boundary of the domain, the resolution of the voxelised input model, and the required simulation time.

Optimised parameters were then used as the basis for a second round of optimisation studies on real living vocal tracts to improve alignment between the ideal parameters and the intended use case of the simulation. These studies initially began on a vocal tract producing the /ɜː/ vowel sound and the vocal tract of the 3000-year-old mummy Nesyamun, 'True of Voice', as a point of interest within this research group. It was quickly determined that optimising for the Nesyamun model was drastically lowering the accuracy on real living tract models due to the specifics of its preservation, and so it was abandoned for three other living tract models.

These studies concluded with a Mean Absolute Error (%) (MAE) across all formants in all four models of 12.77 % with a simulation length of 5 ms, a voxel size of 1 mm, and a normalised acoustic admittance at the domain boundary of 1. These formant frequencies were extracted agnostic to the true values or the model geometry that produced them, and as such it was deemed that an accuracy that was similar to or less than prior research was adequate for this work.

Chapter 5

VTSim: A Complete Vocal Tract Simulation Package

With a satisfactory accuracy achieved across a variety of different models, attention was turned to improving the useability of this simulation routine. One of the goals of this research is that the work contained within will be accessible to a wide audience without a great need for technical knowledge. More specifically, it is not expected that the underlying physics and the specifics of the algorithmic update equations will be immediately understandable to any user, but instead that an average user that has a reason to want to simulate speech sounds from vocal tract models would be able to use this tool without needing any other external input. At this point, using the work shown here so far is a fairly involved process. For ease of development and prototyping, each step of the process is a self-contained Python script, which takes an input based on the previous step of the process and then outputs something useable by the next step.

The first step in modelling the vocal tract in this method is the pre-preparation of the model files into a format that the routine can use. The vocal tract model files that have been provided for this work are in the form of STL files that were designed for 3D printing of the tracts. These files are trimmed so that only the walls of the tract and the air within them are included, from the glottis to the lips. This

trimmed model is then contained inside a cube. The cube is placed such that the lips are on one face and the airway is open to the outside. This produces a cube which has had the airways of the vocal tract hollowed out of it. At this point, the glottis is fully terminated within the cube. An example vocal tract model which has been prepared in this way is shown in Figure 5.1 bisected for visual clarity, although in this particular model the glottis is open to one face of the cube to be discussed further later in this work. This model is then voxelised as discussed in Section 4.3 using an external tool, setting an appropriate resolution by inputting the desired height of the output model in voxels. For example, if the model was 10 cm tall and a resolution of 1 mm was desired then the user would input a height of 100 voxels into the voxelisation tool such that it would produce voxels which have a side length of 1 mm. The voxelisation process produces a file containing the Cartesian co-ordinates of each filled voxel in the simulation domain. One consequence of using STL files for this process is that the STL format optimises the storage of models by only concerning itself with outwardly facing features which define filled volumes. When this file is imported into the next Python script, the voxel grid first needs be 'filled' in the cavity between the vocal tract and the outer surface of the cube. This now filled grid is used to calculate the acoustic admittance values for each grid location in each of the cardinal directions.

Figure 5.1: 'Stern' vocal tract model prepared from acoustic simulation by having its airway enclosed in a cube. This model has been bisected so that the internal airway can be clearly seen.

The array of voxels, representing either tract wall or air, and acoustic admittances between them is then passed into the main body of the routine. This script uses the simulation algorithm discussed in Section 4.2, on a standard rectilinear grid with a generalised boundary update equation based on the locally reacting surfaces model. The simulation inserts a volume velocity source as in Equation 4.16 at a node near the closed glottis and takes a snapshot of the pressure distribution across the entire simulation domain at every time step for the duration of the simulation. These snapshots are loaded into another script that then generates a Vocal Tract Transfer Function (VTTF) using Equation 2.10 at a number of user defined nodes. These VTTF are used to filter a Liljencrants-Fant (LF) model pulse to produce a short speech signal. This speech signal is fed into an Linear Predictive Coding (LPC) algorithm in Praat which produces formant values from the data.

For this research to be easily available to individuals that may be interested

in using it, this process must be made significantly simpler, with as little human intervention required as possible. A prospective user would ideally be able to provide the vocal tract model and then receive formant values after some time, or possibly even synthesised speech sound, throughout the model. As the user would already need access to a vocal tract model it is deemed that requiring them to do some pre-processing, for example in enclosing the tract in a box as described previously, would be acceptable. The simplest way to achieve this is to write another script that deals with running the previously created and discussed ones, and passing information between them. This script would also act as the only user facing part of the simulation, and so should provide good instruction and feedback throughout the process. Figure 5.2 shows all of the current steps required to perform an acoustic simulation using the work in this thesis, with the dashed outline encapsulating the steps that this supervisor script would perform automatically. Creating this script is simple for any of the tasks that are already implemented in Python, but will require more work for any parts of the simulation routine that were previously completely manual. There are three major steps in the simulation process that are entirely manual: the voxelisation of the input model, the choice of nodes at which to input the source signal and measure the pressure, and the calculation of formants using LPC.

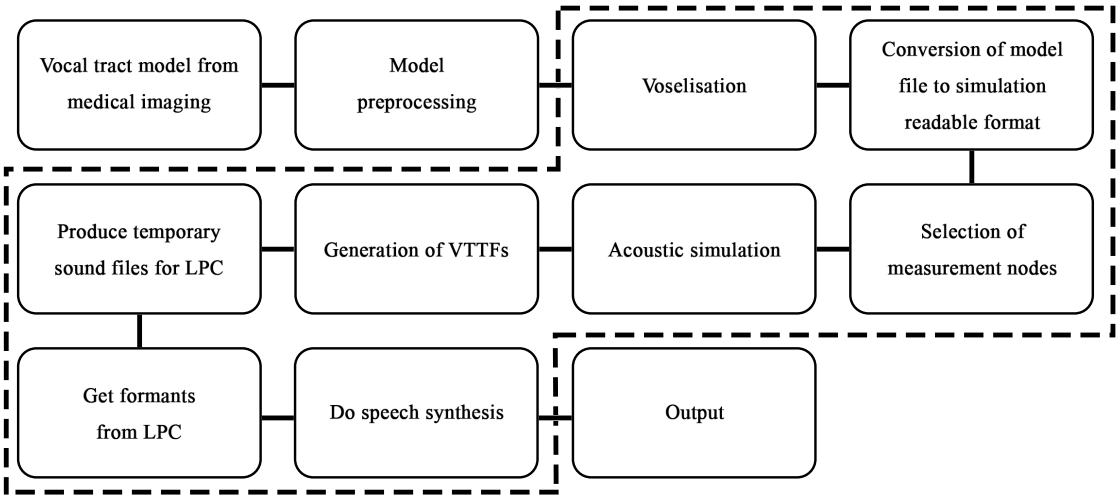


Figure 5.2: All of the steps required to perform the acoustic simulation described throughout this work so far, starting from the top left. The steps encapsulated by the dashed line will be performed automatically by a supervisor script with no human intervention required.

5.1 Automating Manual Processes in Vocal Tract Modelling Routine

There are several pre-existing open source libraries that are capable of voxelisation of STL files in Python. The library implemented here is the Trimesh library maintained by Michael Dawson-Haggerty [113]. Trimesh is a library for loading triangular meshes into Python, with a robust set of built-in features for manipulation and analysis. Trimesh has an emphasis on maintaining watertight surfaces, which was deemed very useful for this simulation to prevent the appearance of any breaks in the walls of the model. After loading an STL into Python using Trimesh, the mesh can be voxelised to the desired resolution using the 'voxelised' command. An approximate comparison of the voxelised model's air volume and the real model's air volume gives a difference of approximately 12.5 % which will vary between vocal tract models. This value is only calculated for one model, and it is expected that the process can also under-estimate the volume by a similar extent. This mesh is then easily converted into a format that matches the other parts of the simulation using another built in command.

Automating the choice of input node and measurement nodes is a much more involved process that will require consideration of what constitutes good choices of nodes in the manual process and how to reverse engineer that decision-making process into some kind of algorithm. The source node was previously selected by choosing the co-ordinate at the centre of the airway at one voxel layer above the lowest layer which contains any portion of the airway. This was to avoid any strange behaviour of input pressure directly next to any vocal tract walls, or from the added glottis termination. This node can be found fairly simply by searching through the voxel grid for that co-ordinate in the same way described. The termination at the glottis within the simulation domain was a simplification made during simulation development and was at this point deemed to be no longer needed. Models would now be open at the surface of the cube, both at the location of the lips and at the glottis. The only empty space surrounding the cube is on the side of the cube where the lips are, so this change will not lead to any erroneous pressure flowing in an unphysical way around the tract rather than through it. With the glottis open against the edge of the simulation domain, it may be that a different normalised acoustic admittance at the surface of the glottis will be required, and so this must be an accepted variable in the simulation routine. The only effect this has on the algorithm described for finding the source node is that that node will now always be on the lowest voxel layer of the domain.

Choice of measurement nodes was done previously by finding the co-ordinate in the centre of the airway at equidistant points along the tract. This required an approximate calculation of the tract length and selection of a total number of points to find. A node was also forced to be just outside the lips, at the edge of the original model bounds before additional propagation space was added in front of the lips. This final node was used as the 'true' output to compare against sound recordings. The geometry of the tract presents significant barriers in the translation of this method into an algorithmic process. Determining the depth of any individual point along the tract is not trivial due to the presence of branches which are occluded

from the direct 'line of sight' of the source node, and the turn made by the tract as it moves into the mouth. By simply taking the length of the vector position of each co-ordinate within the airway, it is likely that some co-ordinates will appear to be closer to the source node than they actually are if only travelling along the airway itself. One method of more accurately measuring this distance would be to perform a flood fill of the airway, storing the co-ordinates that the fill has reached for the first time at each time step. Each point reached for the first time at a given time step would then be at the same distance from the source node as every other point from that same time step. These sets of equidistant points could then have a point selected from them at regular distances to produce the measurement nodes. Some post-processing would still be required however, to ensure that the chosen points are spread linearly along the tract, not directly next to the walls of the tract, and not too close to each other. None of these conditions are strictly necessary for accuracy, but for general use they would make the behaviour of the simulation routine more predictable and comparable across different models.

A similar, but overall faster and more efficient, method was instead implemented for selecting node locations. Beginning from the source node, every co-ordinate in the domain that is within the airway and a user-defined euclidean distance from the source node is found. We can then select a node from this shell and create a new shell around that point to find the next point. This marching spheres algorithm makes use of functions built in to the NumPy array management library in Python, which are heavily optimised when compared to a basic flood fill. Picking the next point from the points within the shell requires some additional logic though, as randomly choosing a point is as likely to move back down the tract as it is to continue along it. Before selecting a point from the shell, all points on that shell that are within a set distance from any previously chosen point are discarded. This distance is larger than the shell radius and is optimised to generally remove all points that are 'behind' the point from which the current shell was drawn. After this filtering process, the average location of all the points left in the shell is calculated and set as the next

node position. This process is then repeated until no new nodes can be found due to all possible new shell points being filtered out. An example of the output of this algorithm can be seen in Figure 5.3. This method has been tested for a variety of vocal tract models and produces, on inspection, good node positions consistently.

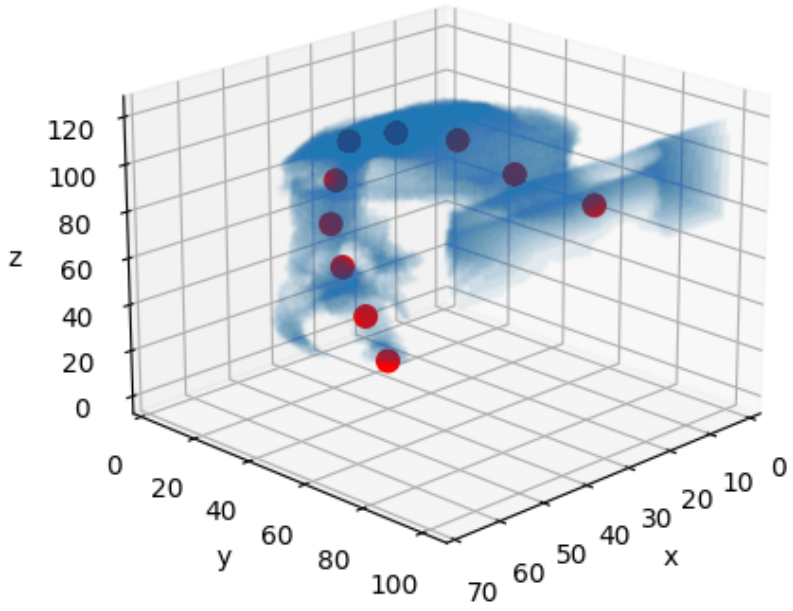


Figure 5.3: Algorithmically generated measurement node locations in 'Stern' vocal tract model. The airway of the tract are shown in blue, and the nodes are plotted in red.

Finally, a method to compute the LPC coefficients of the sound output of the simulation is required. There are several packages which implement Burg's LPC coefficient calculation algorithm into Python, including librosa and LPCTorch [105, 114, 115]. Similar to Praat, both of these packages are Burg implementations, however the formant values outputted from both differ to each other and to Praat. The process of LPC coefficient calculation, as described in Section 4.3.2, is a least squares process that attempts to produce coefficients that satisfy a maximum threshold for the power of the input signal. Depending on the precise implementation of this least squares method could lead to slight variation in the coefficients outputted by different processes. While variation in output does not necessarily infer a lack of accuracy in either librosa or LPCTorch, the value of the ubiquity of Praat in terms of comparison between this research and other publications is high. As such, to

enable direct comparison to other works, it would be ideal to use Praat's formant calculation method itself inside Python.

Parselmouth is a Python library that acts as a Python interface directly to Praat C/C++ code [116]. Compared to a re-implementation of Praat's functionality or an interface layer with Praat's scripting language, Parselmouth will produce identical outputs to Praat in all situations and has access to the full suite of tools available in Praat. This would allow us to also implement voice synthesis based on LPC coefficients through Praat, however it was deemed easier to use a library specifically built for Klatt Synthesis rather than staying within the confines of the Praat workflow [117].

5.2 VTSim Usage and Output Analysis

The VTSim Python script, in its current state, acts as a supervisor that runs the different parts of the simulation routine and handles the passing of data between them. VTSim accepts only one input as a command line argument, which is the path to a file containing a variety of parameters that will control the scripts. Currently, this parameters file must contain the name the user has given to the specific simulation, the path to the STL file that will be used as the model for the simulation domain, and the path to save the outputs of the simulation to. A number of optional parameters are also available, which allow a user to specify the path to a file containing pre-defined node locations for measurement, normalised admittances as domain edges and at the glottis, and the maximum number of formants to calculate at each node. This list is extensible as further functionality is added to VTSim. These parameters govern the running of the scripts that make up the full simulation routine, which are as follows:

- `voxelise.py` - Loads an STL model file into memory using the Trimesh library and performs a voxelisation process on the model, returning an array containing the positions of each voxel.

- `boundaries_calc.py` - Converts the array of voxel positions into a 3D array describing the presence of air or walls at every point, and calculates the admittance between every point in each of the cardinal directions. This version of the script also calculates the position of the source node based on the assumption that the glottis is on the bottom face of the model, but a second unused version of this script is currently present which does not make this assumption and does not calculate the source node.

- `find_nodes.py` - Uses the position of the source node and the 3D array of the model to populate the domain with measurement nodes using a marching spheres algorithm. This script is skipped if a list of nodes is given at runtime.

- `acoustic_simulation.py` - Runs the Wave Based-Digital Waveguide Mesh (W-DWM) acoustic simulation algorithm on the simulation domain. A volume velocity source is inserted at the source node and the entire pressure field in the domain at every time step. Domain boundaries are accounted for using a general locally reacting walls formulation which allows for varying normalised acoustic admittance between the boundary at the glottis and the other boundaries of the model based on the assumption that the glottis is on the bottom face of the model. A second unused version of this script is currently present which does not make this assumption and as such does not allow varying normalised admittance.

- `vttf_gen.py` - Loads the pressure domain at each time step and creates a file for each node location containing the pressure variation at that node at each time step. This script then uses those files to generate a VTTF at each node based on the volume velocity source inserted at the glottis during the acoustic simulation.

- `source_filter.py` - Uses the VTTF at each node as a filter for a LF model pulse to produce temporary speech output for analysis. This script also handles differences between signal lengths by interpolating the shorter data set to have

the same sample rate as the other one.

- `lpc.py` - Uses the Parselmouth library to analyse the temporary speech outputs using a Burg LPC coefficient calculation algorithm to approximate the frequencies of the resonances caused by the vocal tract at each node.

- `speech_synthesis.py` - Uses the `tdklatt.py` module to create speech output for each measurement node from the calculated formants using a Klatt synthesiser.

VTSim provides clear feedback to the user during its runtime, including time estimates and logging of the current stage of the process. Figure 5.4 shows an example log of an instance of VTSim. VTSim will check for prior completions of any of the scripts with a potentially long runtime and will reuse the data from those previous runs if available. In this instance, the acoustic simulation and VTTF generation had already previously been completed, so those steps were bypassed.

```
------ Voxelisation:
------ Voxelisation Complete.
------ Make Boundaries:
Old Domain Shape: (61, 101, 134)
New Domain Shape: (61, 121, 134)
New Domain Size: 989054
Walls before fill: 80542
Walls after fill: 748925
Empty space: 240129
[25, 20, 1]
------ Boundaries Made.
------ nodes.txt Not Found. Generating Nodes:
Nodes found: 16
Redundant propagation space nodes removed: 1
Final number of nodes: 33
Visual confirmation (nodes in red)
Visual confirmation complete.
------ Nodes Generated.
------ Do Acoustic Simulation:
Sim already done.
------ Simulation Done.
------ Generate VTTFs:
Nodes to make: []
Load node: nodevalues(26, 12, 19).npy
Load node: nodevalues(27, 10, 39).npy
Load node: nodevalues(25, 9, 59).npy
Load node: nodevalues(27, 7, 79).npy
Load node: nodevalues(29, 14, 98).npy
Load node: nodevalues(31, 30, 110).npy
Load node: nodevalues(32, 49, 115).npy
Load node: nodevalues(28, 68, 118).npy
Load node: nodevalues(32, 86, 111).npy
Load node: nodevalues(31, 101, 111).npy
------ VTTFs Made.
------ Do Temporary Speech Signal Generation:
------ Temporary Speech Signal Generation Done.
------ Calculate Formants:
Node Position: (26, 12, 19)
Formants: [ 775 1953 2869 3752 5380]
Bandwidths: [ 556  519  647  616 1221]
Node Position: (27, 10, 39)
Formants: [ 788 1964 2894 3768 5287]
Bandwidths: [ 597  533  654  636 1230]
Node Position: (25, 9, 59)
Formants: [ 807 1977 2923 3793 5089]
Bandwidths: [ 655  558  672  686 1197]
Node Position: (27, 7, 79)
Formants: [ 807 1979 2941 3805 4717]
Bandwidths: [688 565 632 700 572]
Node Position: (29, 14, 98)
Formants: [ 774 2037 3045 4018 4891]
Bandwidths: [681 598 607 748 714]
Node Position: (31, 30, 110)
Formants: [ 771 2055 3069 4068 4936]
Bandwidths: [697 619 602 728 729]
Node Position: (32, 49, 115)
Formants: [ 783 2068 3083 4094 4955]
Bandwidths: [769 656 613 730 744]
Node Position: (28, 68, 118)
Formants: [ 821 2082 3095 4118 4965]
Bandwidths: [959 740 651 762 786]
Node Position: (32, 86, 111)
Formants: [ 718 2081 3101 4113 4975]
Bandwidths: [529 596 548 642 668]
Node Position: (31, 101, 111)
Formants: [ 648 1909 2995 4023 4937]
Bandwidths: [197 479 543 661 662]
------ Formants Found.
------ Do Klatt Synthesis:
------ Klatt Synthesis Done.
------ Cleanup.
------ Modelling Routine Complete. ------
```

Figure 5.4: Example runtime output log of VTSim package.

The VTSim package has been used to perform an additional set of simulations

for the purpose of calibration against the separately run scripts. This was deemed to be necessary as several small changes were made to the implementation of the various scripts so that they could easily pass data between them, which could have introduced small errors or variations. The most major of these changes was in the rewriting of the acoustic propagation algorithm function to allow for user inputted normalised admittances at the glottis and the rest of the bounds. In addition, while it is not a change made to implementation scripts per se, the process by which voxelisation is performed throughout this work has now been abandoned in favour of the Python implementation given by Trimesh. While all voxelisations at the same resolution are still approximating the same original model and should produce equivalent results, there is likely implementation differences or even variation in voxelisation algorithms that may lead to a change in accuracy. This is even more likely given the previously discussed heavy dependency of the output formant accuracy on the model geometry.

As can be seen in Table 5.1, the average accuracy of the produced formants using VTSim is 0.85 % higher than those produced from previous simulations. This may not be statistically significant, however the standard deviations of the mean absolute errors across the manual and VTSim data are 3.847 and 2.691 respectively. This shows that the real gains in accuracy in the process of translating the process into a single package is in a significantly lower variance in errors. There are two major potential causes for this change. The first possibility is that the Trimesh voxelisation algorithm provides more accurate representations of the input model than the previous method. Unfortunately, the precise algorithm used for the previous voxelisation method is not made clear, so any differences are purely speculation. Secondly and most likely, improvements in accuracy could arise from the changes made to the various scripts in terms of fixing small errors or possibly in minor implementation changes.

Table 5.1: Percentage error on individual formants and Mean Absolute Error (%) (MAE) across all formants, compared to audio recordings, from acoustic propagation simulations. Data shown covers a variety of vocal tract models articulated to produce the given vowel sound. Individual formant errors and MAE are given for the 'Manual' W-DWM simulation method discussed in Chapter 4 and for the automated 'VTSim' package described in Chapter 5. Blank cells appear when the Praat LPC algorithm found only four formants for a given output.

Simulation Details		Percentage Formant Error (Hz)					Mean Absolute Error
		F1	F2	F3	F4	F5	
Manual	/iː/	-24.79	0.07	-22.66	-5.53	2.18	**11.04**
	/uː/	-23.73	-5.02	-30.27	-5.46	1.46	**13.19**
	/ɑː/	-5.26	46.42	16.50	0.44	-	**17.16**
	/ɜː/	3.68	-8.73	-10.99	-2.60	-	**6.50**
VTSim	/iː/	-25.03	0.46	-22.18	-5.11	2.35	**11.02**
	/uː/	-13.44	2.31	-29.19	-2.90	1.83	**9.93**
	/ɑː/	-10.37	36.55	12.85	-1.59	-	**15.34**
	/ɜː/	9.96	-9.35	-11.34	-1.35	-	**8.00**
Average Manual		14.36	15.06	20.11	3.51	1.82	**10.97**
Average VTSim		14.70	12.17	18.89	2.74	2.09	**10.12**

The accuracy level of the current full routine of acoustic simulation is still similar to if not better than previous works, which is deemed adequate to produce useful scientific output.

5.2.1 Comparing VTSim Outputs to Simplified Examples

While the VTsim outputs presented above seem to be accurate when compared to physically recorded values, it is difficult to discern the robustness of this comparison. This thesis presents comparisons between ten simulated vocal tract models and their respective physical audio recordings. Previous work has reported similar

comparisons between only six models, often fewer or no real recording comparison at all [4, 112]. Approximate formant frequencies for particular speech sounds reported on in literature will often vary slightly from an individuals formant frequencies, so using only values from literature will always impart some error in the analysis. Testing simulation outputs against real physical audio recording should avoid this error contribution, however the act of performing the measurements also likely introduces its own amount of error so this is not a perfect solution to the verification problem.

A more robust method of simulation verification, but one that is less directly related to the environment in which this model is designed for, is in testing the simulation routine against simple models that can be solved exactly. Equation 2.9 presented the method for calculating the frequencies of allowed standing waves in an tube that is closed at one end and open at the other. By using such a model as the input to the simulation, the simulation outputs can be directly compared against known exact formant frequencies to assess its ability to simulate acoustic propagation in this setting. A simple tube model created for performing this simulation can be seen in Figure 5.5 and the output of the voxelisation of that model can be seen in Figure 5.6.

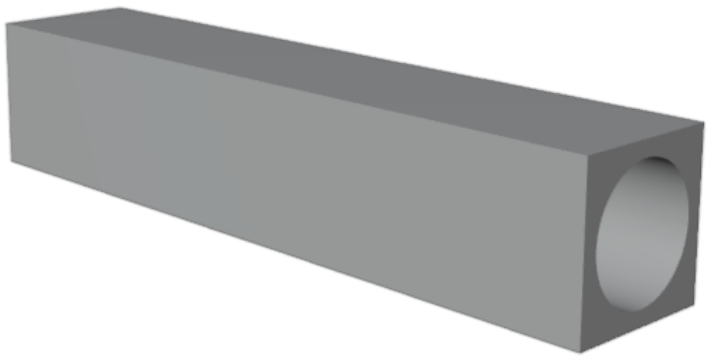


Figure 5.5: Input model for simplified model verification.

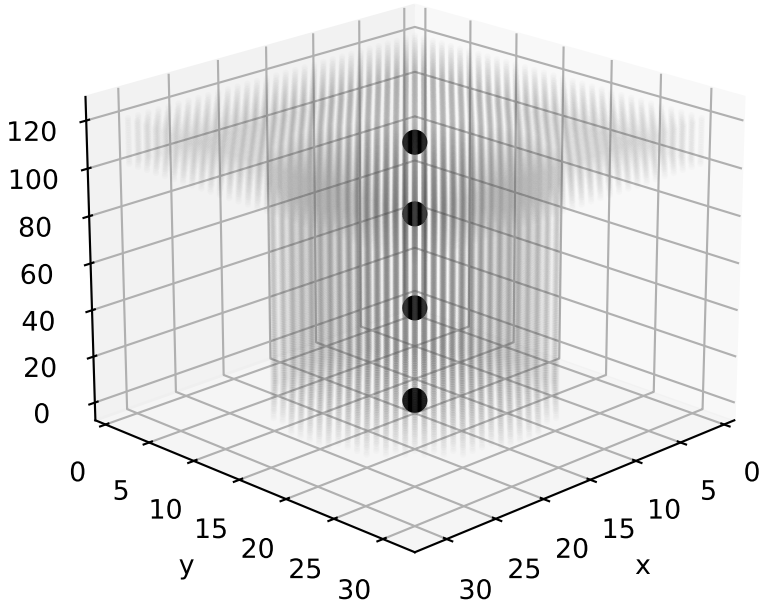


Figure 5.6: Voxelisation of simplified model. Air is shown by the shaded area and the nodes at which the formant measurements are taken are shown in black. A small region of air is added at the end of the tube, the top of this figure, for acoustic propagation.

Table 5.2 shows the percentage error in formants measured along the centre line of two tubes with constant cross-sectional area. The tubes used in this simulation both had a radius of 1.2 cm, but other radii produced very similar formant frequencies when tube length was kept constant. Average absolute error across all five formants for both models is approximately 7.4 %. A few conclusions can be drawn from this data, firstly that the simulation routine does not perfectly recreate a simple example, but does produce formants which are more accurate than those produced for significantly more complex models as shown in Table 5.1. The inaccuracy in the simplified model, as discussed throughout this work and to be discussed more thoroughly in Chapter 6, could be due to the optimisation of the simulation routine for 3D vocal tract models. The optimisation processes that have been conducted in this work have had the goal of aligning simulation outputs to physical audio recordings. As the simulation routine has been shown to be sensitive to parameters such as domain boundary reflections and resolution, those parameters have been

finely tuned to produce the most accurate outputs in the intended use case of this routine. It is possible that the values those parameters have taken are a good fit for the larger, more cubic, domain that a vocal tract model occupies and its more complex geometry.

Table 5.2: Average error in formant frequencies along the centre line of tubes of uniform cross-sectional area.

Tube Length	Percentage Formant Error					MAE
	F1	F2	F3	F4	F5	
7 cm	8.279	-3.773	-12.757	-11.867	0.472	**7.429**
14 cm	4.715	-1.783	-10.441	-9.267	-10.899	**7.421**

Also of note is the distribution of errors in the formants. F1 and F5 had the most variation in percentage formant error, with the other formants having a similar magnitude of error. It is difficult to tell if this variation is statistically significant however. The standard deviation across all of the formants in this study is 4.12, with the 7 cm model formant errors having a lower variance than the 14 cm dataset. It is likely that this variation is not statistically significant, however verifying this would require a larger breadth of simulations, which are not possible as part of this work due to time restrictions. With more time, a much larger study of simple constant radius tubes, alongside other common examples such as two connected tubes of varying length and radius, would be performed to better interrogate the successes and failings of this simulation routine.

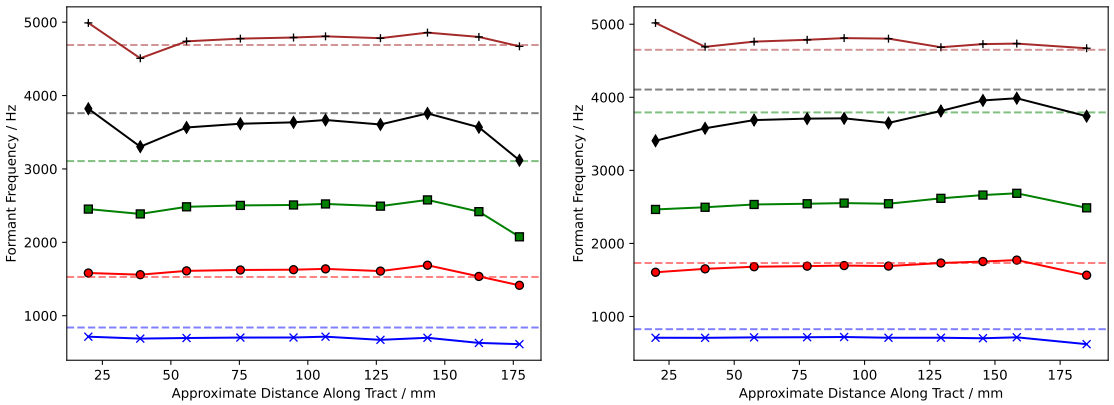
As a final comment on the validity of these simulations, these simplified models are not fully equivalent to the standing-waves-formed-in-a-tube example when viewed by the simulation routine. As this method was designed to include propagation within the walls of the vocal tract, that behaviour is inseparable from the update equations of the simulation algorithm. This means that propagation is being modelled in the walls surrounding the simple tube as though they were the walls of the vocal tract. This also comes with the inclusion of domain boundary reflec-

tions as those walls meet the edge of the simulation domain, and some free space at the end of the tube. Propagation in the walls may affect the formant frequencies produced by the simulation in a complex way which would affect the ground truth formant frequencies of this system. This simulation routine is not a general acoustic simulation tool, it is designed purposefully to reproduce the acoustic propagation in 3D vocal tract models. While it would be preferable that a full study of accuracy on simple examples would be performed, this simulation routine is not a perfect match to the physics of those examples.

Despite some mismatches in the underlying physical situation, as with the Arai models these simulations do remove a large number of the 'degrees of freedom' from this simulation process. There is still complexity which makes it difficult to assess the usefulness of these validations. It is expected that the variations in the ground truth induced by these complexities should not be large though, so these simulations do still show that this routine can produce formant frequencies that agree with mathematically calculable ones to a reasonable extent and add to the proof of accuracy of this model.

5.2.2 Acoustic Profiling Along the Length of the Vocal Tract

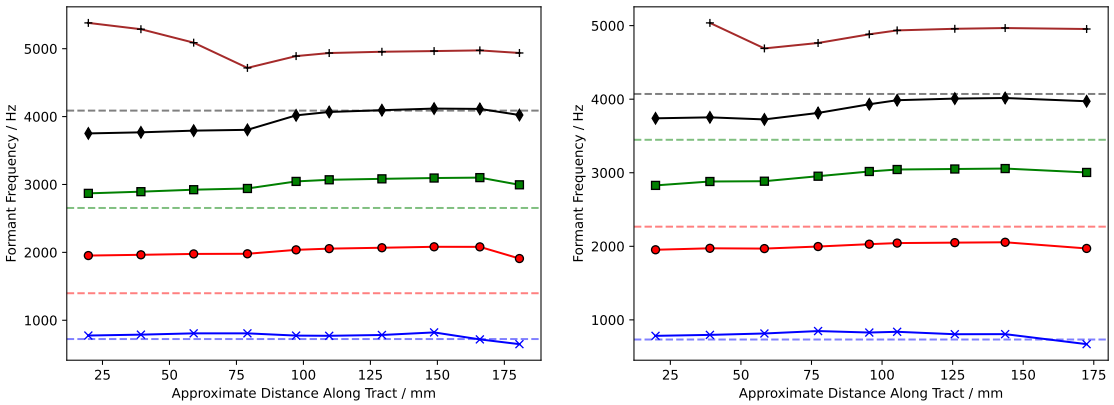
Figures 5.7 and 5.8 show the variation of formant frequency along the length of the tract, from just above the glottis to just outside the lips. In each figure, the formant frequency extracted from the physical recording is shown as a constant frequency line for each formant. The differences between the formant frequency at the furthest distance along the tract and the constant frequency value for each formant is a visual representation of the given average errors discussed so far. Apart from acting as a visual aid to show the extent to which the final values of each formant are under or overestimated based on their physical counterparts, the recorded formant values should be disregarded in favour of examining the relative variation of each individual formant along the tract.



(a) 'Neap' model formants.

(b) 'Food' model formants.

Figure 5.7: Extracted formant frequencies at increasing distance along the tract from the glottis. The colours blue, red, green, black, and brown show the formants from the first to the fifth in increasing order. Constant frequency lines show the formant frequencies extracted from the physical recording of the corresponding tract, for comparison with the formant values at the furthest extent.



(a) 'Hard' model formants.

(b) 'Stern' model formants.

Figure 5.8: Extracted formant frequencies at increasing distance along the tract from the glottis. The colours blue, red, green, black, and brown show the formants from the first to the fifth in increasing order. Constant frequency lines show the formant frequencies extracted from the physical recording of the corresponding tract, for comparison with the formant values at the furthest extent. In the physical recordings for both of these vocal tract models, a fifth formant frequency was not obtained from the formant calculation algorithm.

The speech formant is a characteristic of the effect of the vocal tract on an input. As discussed throughout this work, a speech sound is made up of a number of formants with different frequencies. As a resultant characteristic the frequency

of the formants should be fixed, especially in a for a time-invariant system. Figures 5.7 and 5.8 show apparent frequency variation within the tract however.

While formant frequency should be fixed, the amplitude of different frequency components will likely vary at different locations depending on which frequency components are attenuated by the resonances of that region. As formant frequencies are extracted from the peaks in the power spectral envelope, local variations in amplitude could lead to the formant calculation algorithm slightly shifting its calculated formant frequency away from the resultant values. As such, the variations seen in the figures may show the locations in the tract that have higher or lower frequency resonances than the resultant formant frequencies.

These figures generally show that the extracted frequencies of higher frequency formants are more greatly effected by changes in geometry, showing greater shifts in frequency along the course of the tract. This is likely due to a generally lower magnitude of the power spectral envelope at higher frequencies that is more sensitive to local changes in amplitude. Across all models a general increase in frequency in all extracted formants is observed in the first 150 mm followed by a dip in pitch over the last 30 mm. It should be noted that the distances described are approximate and only directly related to the scale of the simulated tract rather than the true scale of the tract, and should not be taken as accurate to the scale of a real human vocal tract. By comparing the data to the tract model itself, some suggestions of the source of these frequency shifts can be made intuitively.

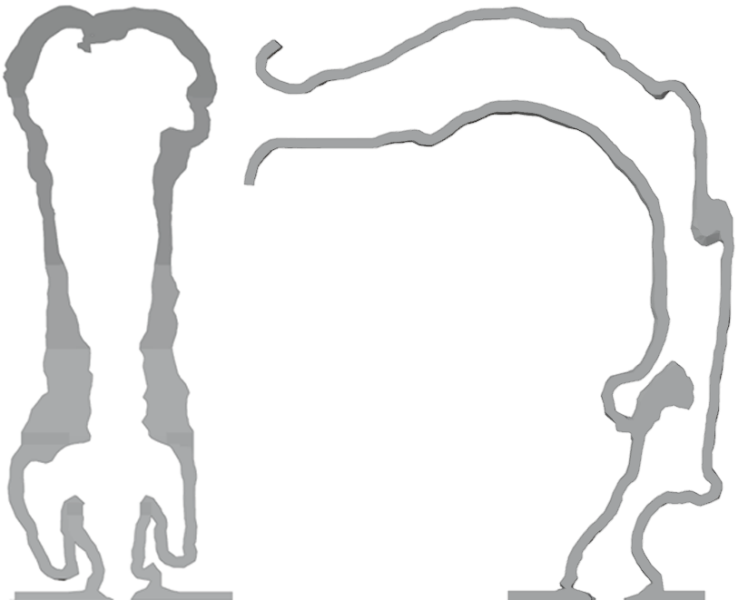


Figure 5.9: Cross section of 'Stern' vocal tract. Left: In the frontal direction along the mid-sagittal line. Right: In the median direction along the mid-sagittal line.

An acoustic pressure wave is produced at the vocal folds when they are vibrated while under tension. This vibration will have a fundamental frequency and harmonic frequencies, which are all of the integer multiples of the fundamental frequency. This sum of frequency components then passes into the vocal tract and is filtered by it.

Figure 5.9 shows cross-sections of the vocal tract model corresponding to the simulation data of the /ɜː/ vowel sound. Some detail is lost in these views, but it can be seen that generally the tract at the larynx is quite constricted, with several narrow side passages for example the piriform fossae, and becomes much more open as it moves into the mouth. Any region of space will have a particular acoustic resonance frequency based on its dimensions. In a simple geometry like an open tube, the lowest resonant frequency of that space is given as approximately $v/2(L+1.2r)$ where v is the speed of sound, L is the length of the tube, and r is the radius of the tube. Resonance is a passive phenomenon in which power repeatedly added to a system reinforces power that was previously added to that system due to synchronised reflection of, in this case, an acoustic wave. If the power being added into a system is added with a frequency similar to the fundamental resonant frequency of that space, or one of its harmonic frequencies, then it will be amplified. The other acoustic effect

in the tract relevant to the resultant sound is that of acoustic impedance: as sound travels through an impeding medium, it will progressively lose energy until it is fully attenuated. This effect is often frequency dependent, but for the purposes of this analogy it can be taken to be frequency independent. Impedance of the air itself is also not the only effect that causes a loss in acoustic power during propagation, for example when the wave reflects off of a surface it will also generally lose power. These effects of amplification and power loss combine to act as a filter for the input from the vocal folds.

When the sum of frequency components travelling from the vocal folds propagates into a region of the tract, if one of the vibrating components is similar to the resonant frequency of that space then air in that space will begin to resonate with that component, leading to a lack of attenuation of that particular frequency component over time while the other components continue to be attenuated by the acoustic impedance of the medium. This amplification comes from the effective collection of energy in the resonant space over the process of multiple 'injections' of energy from the vocal folds.

Two tubes with different lengths and radii will have different fundamental frequencies. The input signal from the vocal folds repeatedly propagating through one tube will experience amplification of frequency components similar to the fundamental frequency of the space or its harmonics, and attenuation of other components. As that propagating field moves from the first tube into the second tube, it will now begin to experience amplification at different frequencies based on the fundamental frequency of the new tube, and attenuation at others. If the new tube has a larger radius or length, then it will also have a lower fundamental frequency. As such there will be amplification of frequency components lower than that which was previously amplified. Conversely, if the wave propagates into a tube with a smaller radius or length, higher frequency components will be amplified by the repeated power input from the vocal folds.

So while the frequencies of the peaks of the power spectral envelope will quickly

settle into the frequencies dictated by the geometry of the system, the amplitudes of frequency components in different tube segments may vary based on their geometry. If we now relate this back to the behaviour we see in Figures 5.7 and 5.8, we can identify locations along the tract that might have significantly different resonant properties than the tract as a whole. If the speaker is attempting to vary their speech formants in a particular way, they could target change at regions of the tract that show frequency shifts in the undesired direction.

As a speculative example, in Figure 5.7a we observe that the extracted formant frequencies at the lips seem to decrease dramatically. This would imply that the area just before the lips has a lower resonant frequency, and raising the frequency of the speech formants may be done most effectively by constricting that region specifically.

This is still very hypothetical, but could allow for a more data driven approach to speech therapy or instruction. Other potential uses of direct access to the way in which the geometry of the tract affects the speech it produces can be found in fields such as teaching or reconstructive surgery. If an individual was undergoing some kind of surgery within the tract for medical reasons, a surgeon could use a tool based on this research to try to align any reconstruction to formants extracted from that individuals recorded voice, and could potentially target those changes to the most effective parts of the tract. Further still, the individual could possibly even request that parts of their tract are changed to modify the natural formants in a way that may be beneficial to their singing voice part or to their speech projection for public performance to better match the space they usually perform in. This research could provide an important first step in informing the development of these potential applications.

Existing research has explored the effect of artificially removing parts of the vocal tract to investigate their effects on formant production, however the research has typically been focussed on changes in the magnitude of resonances rather than their frequencies, and has also only used either physical 3D printed tract models or

simplified electronic models [118, 119]. Figure 5.10 shows the /ɜː/ vowel sound vocal tract both fully intact and with the piriform fossae removed. By simulating acoustic flow through both models and comparing the outputs, we can see a quantitative measure of the effect on speech that the piriform fossae has.

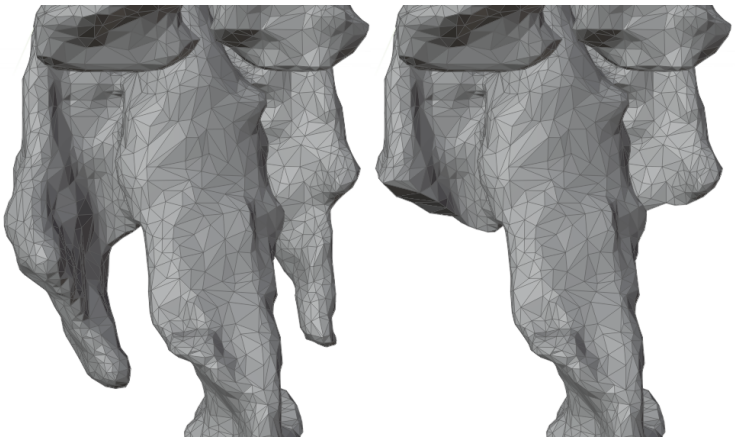


Figure 5.10: Vocal tract model of 'Stern'. Left: Original tract. Right: Tract with piriform fossae removed.

The average variation in extracted formant frequency, across all formants, along the tract as a result of the removal of the piriform fossae is shown in Figure 5.11. As would be expected from the open tube analogy presented above, the average extracted formant frequency decreases early in the tract due to the removal of a tightly constricted chamber within which the resonant frequency would be high. The same analogy does not easily extend to the behaviour seen in the rest of the tract though, with a steady increase in average extracted formant frequency along the tract followed by a sharp drop in frequency around the mouth.

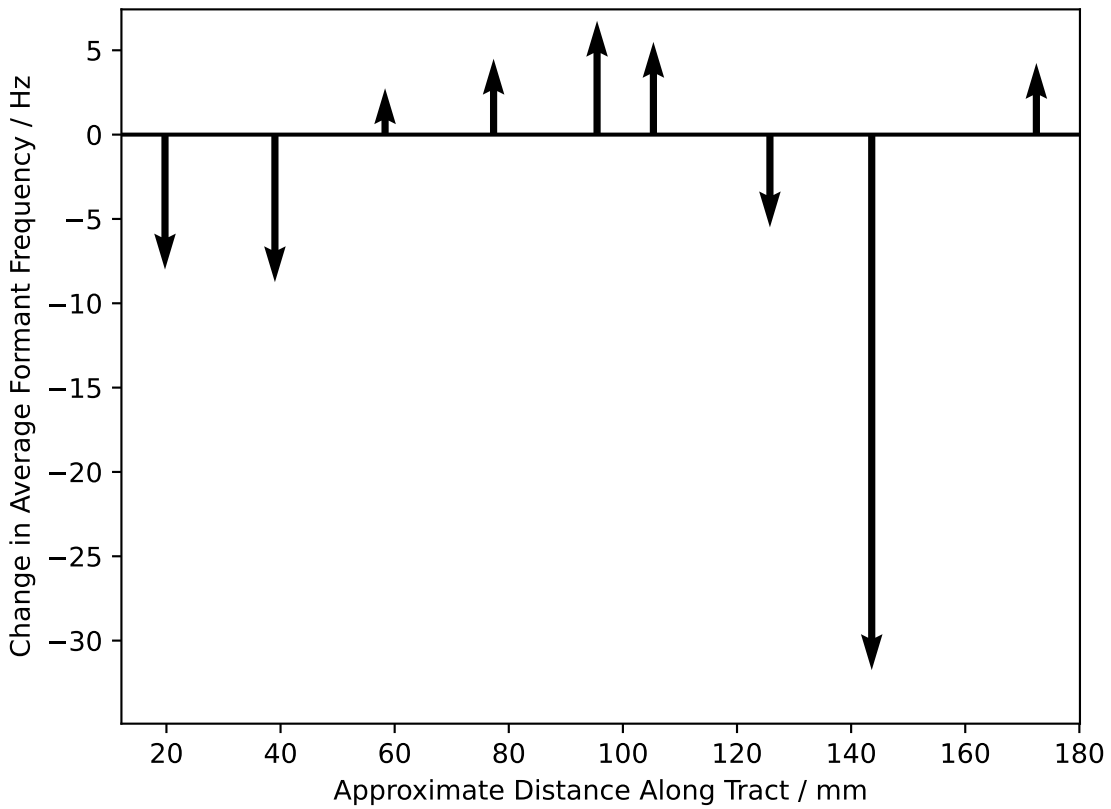


Figure 5.11: Average shift in extracted formant frequency across all formants along the 'Stern' vocal tract when the piriform fossae is removed.

The ability to interrogate the behaviour of the tract both along it, and under the effects of geometrical manipulation, in an intuitive visual way using a software package which is designed with usability in mind is one of the main outcomes of this research. This procedure and analysis represents novelty in the field and has the ability to be applied to a wide variety of future research and real world applications. While the VTSim package and its outputs have been validated to an acceptable degree of accuracy based on comparison with prior research, that validation was only based on recreating the outputs of physical vocal tracts in simulation from the same inputs. While this author believes that the validity of the output at the lips does suggest validity of the data within the rest of the model, providing some direct quantitative insight would greatly improve the degree to which these analyses are considered trustworthy enough to influence real world applications.

5.3 Chapter Summary

The creation and validation of an all-in-one vocal tract acoustic simulation package is described throughout this chapter. A discussion of the processes that make up the previous simulation routine identifies manual processes that are then automated directly in Python, allowing for the simulation to accept a 3D model file and a set of parameters and produce formant frequencies and synthesised speech sounds from them. This is the first example of a complete acoustic simulation package which is designed specifically for vocal tract acoustics that is known of at time of writing. The package is available at this address: https://github.com/dotdandotunderscore/VTSim.

VTSim is shown to produce formants across four real physical models with a MAE of 10.12 % and a standard deviation across the MAEs of each model's formants of 2.691. This is a gain in accuracy over the previous simulation process and well within the accuracy stated by prior research by Gully et. al [4, 112], with the benefit of being compared to real physical outputs across the first five formants. VTSim was also used to perform simulations of simplified acoustic examples with a MAE across both simulations of 7.425 % and a standard deviation of 4.12.

This simulation package was then used to perform simulations of the evolution of the acoustic field within the vocal tract to explore the way in which speech arises based on the geometry of the tract. Changes in extracted formant frequency along the tract were related to the geometry using simple physics concepts, showing the potential power of this approach. A similar simulation and explanation was performed on a vocal tract which had its piriform fossae removed to investigate the effect of geometrical changes on speech production. It is suggested that the accuracy of the formant reproduction at the lips is its own validation of the accuracy of propagation data within the model: For accurate sound to be produced at the lips, the acoustic propagation throughout the whole domain must be somewhat accurate as that propagation is what produces the output at the lips. It is still important however to attempt to prove this through measurement.

Chapter 6

Acoustic Profiling in the Tract Using Physical Measurements

While the data shown in Section 5.2.2, and the conclusions drawn from it, are reasonably well validated by the accuracy of the outputs of the simulation, there are concerns over whether the ability of the simulation routine to produce an accurate output at the lips does imply accuracy of the simulation of acoustic propagation data within the vocal tract.

To be able to confidently assert that the simulation outputs within the vocal tract are accurate enough to provide good insight to speech production, direct comparison between simulation outputs and measurements from a physical model would be ideal. This chapter will cover the experimental design and procedure of a set of measurements which have been performed to help further validate the simulation routine.

6.1 Measuring Acoustic Field Within the Vocal Tract

Measurements will follow much of the same procedure used for other measurements presented in this thesis. The basic experimental plan first involves producing the

vocal tract for measurement via 3D printing. Vocal tract geometry data originates from the work by Speed et al. [88] and have been converted to 3D model files. The files undergo a large amount of preprocessing to remove geometric artefacts which would cause printing errors, for example internal surfaces and non-manifold geometry.

The model is given some amount of wall thickness which has been tuned in prior works by this group for subjective similarity between sound played back through the model and the original audio recordings made during the 3D scans. The thickness used in this work, 2 mm, is considered as 'conventional wisdom' within this research group and has not been tested as part of this research. Notably, this thickness tuning was done based on tracts produced by printers and in materials which may not match the properties of the ones used today. As such this thickness may not be as accurate as it was previously considered to be, but testing this would require access to the previously used printers and materials which is unfeasible at this time.

A coupler is then added to the bottom of the vocal tract which will act as a stand and form the seal between the tract and the loudspeaker that it will be placed upon. The loudspeaker shaft is 34.5 mm in diameter and 18 mm in depth. The coupler is designed to be 0.5 mm larger in both of those dimensions to ensure the tract will fit on the loudspeaker. Electrical tape is used to add small amounts of thickness to the loudspeaker shaft after printing if the seal between the tract and the loudspeaker is deemed to be too loose. All tracts are printed from PLA on a Stratasys F170 multi-material printer [120]. This printer uses a soluble support material to ensure that the internal geometry of the tract does not collapse during printing, and is dissolved after the print is complete. The stated resolution accuracy of the printer is 0.2 mm which should be sufficient to resolve most if not all the geometric detail, and is five times larger than the resolution of the acoustic simulation routine.

The loudspeaker is attached to the Vocal Tract Organ produced by Howard [111], which deals with the production of a sound source to play from the loudspeaker which is accurate to the sound produced at the glottis in the real vocal tract. The

Vocal Tract Organ connects directly to a MIDI controller, in this case a keyboard, to control activation and deactivation of the output, and the fundamental frequency of the sound source. In previous measurements, a microphone has been held between the lips of the model during sound production to capture a few seconds of output which are then used for analysis.

The major variations in experimental design arise in the consideration of how to perform internal measurements within the tract. In an ideal situation, the measurement would be fully unobtrusive and have no effect on the sound output of the model. This is unfortunately not possible with measurement devices of finite size. The measurements also need to be conducted from the exact location that the acoustic model outputs its data if possible. The simulation operates with a much lower resolution than the physical models, so this may also prove complicated. Finally, the simulation outputs the data directly as it is produced, but real microphones are often designed with some non-linear frequency response based on their planned application. This frequency response would lead to a change in amplitude of different frequency ranges which could affect the results of the measurement.

The only option that was produced which at least partly satisfies the above conditions is to add holes in the walls of the vocal tract that a miniature microphone could be lowered into such that they would be flush with the vocal tract wall around them. The microphone chosen for these measurements was the DPA Microphones' 4060 Series Miniature Omnidirectional Microphone [121]. This microphone has an outer casing diameter of 5.4 mm and an actual sensor diameter of 4.75 mm. This is small enough that it can be fit into the model in locations in which the walls of the tract were already relatively flat which should help to minimise the microphones effect on the propagation of sound. The microphone also has an exceptionally flat frequency response between 30 Hz to 6000 Hz which is sufficient to capture the first five formants with minimal amplitude change.

Figures 6.1 and 6.2 show the hole made in the model and the assembly which will be attached to it respectively. A hole which is slightly larger than the sensor

diameter of the microphone is added to the model in locations which are close to, and point towards the locations of, each of the nodes that the simulation routine outputs data at. The minimum amount of material is removed from the walls such that the microphone will remain relatively flush with the inner wall of the tract. In early prototypes of this design, it was noticed that ensuring the microphone was accurately and consistently oriented was close to impossible and so a guide was deemed necessary to be attached to the outside of the tract to hold the microphone in position. The square cut out around each of the microphone holes on the vocal tract was added to provide a surface to mount the microphone guides onto, taking care to remove as little material as possible to not greatly affect the acoustic properties of the walls. The guides were attached to the model using a strong adhesive which has the benefit of filling any unwanted air gaps between the guide and the tract wall. This adhesive likely has different acoustic properties to the printed material however and may still influence the acoustic propagation. Plugs were also created to fill the holes which were not currently being used for measurements, which could be added and removed with ease. These plugs were designed to try to sit flush with the inner wall of the tract, however this was generally approximate and subject to any small offsets introduced during the printing process.

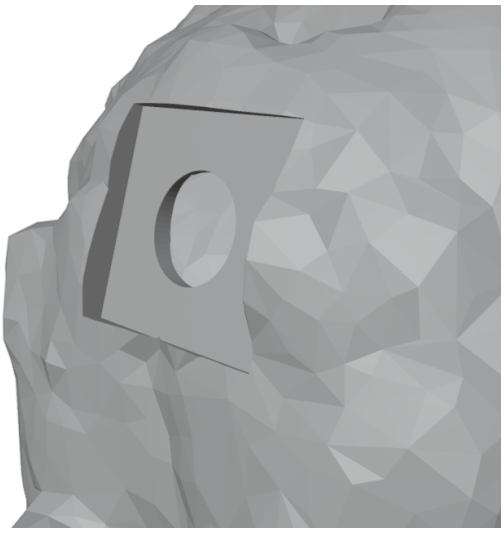


Figure 6.1: Image of microphone hole and surrounding cut-out in the back of a vocal tract model.

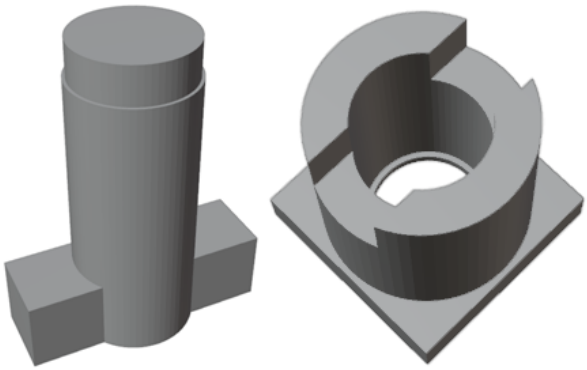


Figure 6.2: Image of the guide to be attached to the cut-out shown in Figure 6.1 and the plug which will be fit into the guide when it is not in use.

Figure 6.3 shows the printed model prepared for a measurement. Some electrical tape was added to the plugs to improve the seal between them and the guides, and also to stop them from rattling during sound production. Measurements were made through an audio interface directly into audio recording software on a single audio channel. Measurements were performed in a room with low enough background audio that the microphone did not detect any sound.

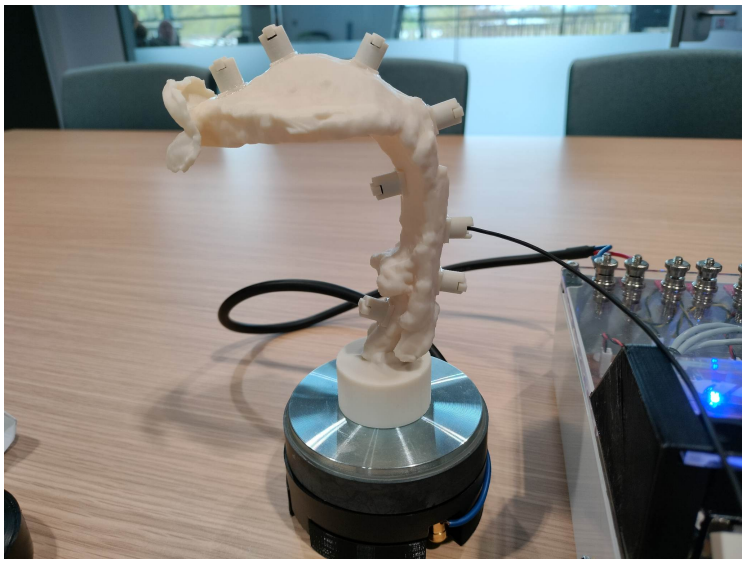


Figure 6.3: Photograph of 3D printed vocal tract model with microphone guides and plugs fitted. The microphone is currently inserted into a guide on the right side of the image.

It is important to recognise that while much greater care has been made during this series of measurements to minimise external factors that could affect the data, due to an assumption that the internal measurements could be more sensitive to

external factors than the measurements at the lips, there are still many factors that may cause inaccuracies. The first of which, which has been mentioned several times before, is that the 3D printing process has finite resolution. Depending on the orientation of the model when it is loaded into the printer, the exact orientation of the print layers will vary which may occasionally lead to inaccurate representations of the geometry. This is especially relevant when considering the newly added microphone guides and holes. The way that the guide attaches to the model, and as such the way that the microphone sits when it is inserted, is very sensitive to the geometry of the surface between the two pieces. At certain locations it is possible that the microphone protrudes into the model enough that it may have an effect on propagation, and at others the microphone may not protrude enough and could be slightly occluded. As the output nodes in the simulation are designed to be as far from the vocal tract walls as possible to minimise any unexpected reflection artefacts, it is also impossible to take physical measurements in the exact same locations.

6.2 Comparison of Physical Measurements and Simulation Outputs

Before performing the full internal measurement, a series of characterisation measurements were needed to properly understand the sensitivity and properties of the measurement process. Table 6.1 shows formant frequencies measured from one vocal tract model when varying the input frequency by 10 Hz to 20 Hz. If the vocal tract is a Linear Time-Invariant (LTI) filter, which these static 3D printed tracts should be over a small enough sample period, then the formant frequencies would be independent of the input frequency. This independence is unfortunately not possible to observe in a real analysis process.

As discussed in Section 5.2.2 the source signal produced at the vocal folds is effectively filtered by the amplification of frequency components by the resonant spaces within the tract, and by acoustic power losses suffered as it propagates through the

tract. The key issue with acoustic analysis is that, as a passive system, the vocal tract only effects the amplitude of the original frequency components. A space in the tract may not have a resonant frequency that exactly matches one of the harmonic frequencies of the source but will still produce amplification of that harmonic. As such the true formant frequencies, which correspond directly to interactions between the source and the vocal tract, may not be present as frequency components in the output sound. As the output only contains energy at the original fundamental and harmonic frequencies, analysis methods that are attempting to obtain the actual formant frequencies cannot simply return the frequencies in the output sound which have amplitudes above some particular amplitude. LPC attempts to do this by using approximate formant frequencies to reproduce a source signal. The issue of formant frequencies not falling precisely at the frequency of the harmonics is itself the solution that LPC leverages: An individual formant frequency may cause the amplification of multiple harmonic frequencies if they are similar enough in frequency to that formant. This can provide multiple data points from which to fit the effect of a formant frequency to and enable a better approximation to it. This is effective if the fundamental frequency of the source is low enough, so as to have harmonic frequencies that remain relatively similar in frequency up to frequencies above that which are particularly important for speech intelligibility. As such, LPC will generally be more accurate for male speech than female speech due to a lower f0.

This inaccuracy is built in to LPC and as such different values of f0 used in the physical recordings will likely effect the other formant frequencies. If this effect leads to too large a loss of accuracy, then special care will need to be taken to ensure f0 is similar in the recordings and the simulations, if not choose f0 carefully to minimise the effect.

As can be seen in the table, there is slight variation across the frequency range leading to a maximum MAE over these measurements of 1.40 %. There are a few possible explanations for this observation. The first is that the room these mea-

surements were performed in could have some resonances or anti-resonances which are excited at frequencies close to the input frequency. This would act as a second filter whose characteristics could affect the frequencies of the extracted formants. A second option is that discussed prior, of the inherent inaccuracy of analysis methods such as, but not limited to, the LPC algorithm used here in only trying to minimise the power of any residuals present in its assumed input signal when applying the inverse filter to the data. While the simulation algorithm is not affected by f0, analysis of its results likely is. The most troubling possible explanation is that these tracts are not truly LTI filters and could themselves have some resonance that is coupling to the input frequency and causing variation in the formants. To minimise any errors resulting from the input frequency, it will be fixed at 131.8 Hz for future measurements and when producing simulation outputs. This specific value has no physical underpinning, apart from being in a sensible frequency range for human speech and as it was the default setting on the synthesiser used here.

Table 6.1: Formant frequencies measured at the lips of a 3D printed /ɜː/ tract when varying the input frequency. 131.8 Hz was used here as the control frequency from which to compare other values when calculating the Mean Absolute Error (%) (MAE).

Input Frequency	Formant Frequency (Hz)					MAE
	F1	F2	F3	F4	F5	
131.8 Hz	708	1318	2942	3951	5441	-
120.1 Hz	702	1315	2868	3941	5269	**1.40**
150.2 Hz	714	1301	2956	3959	5326	**0.99**

An identical set of measurements was performed whilst varying the amount of gain added by the Vocal Tract Organ to the sound source played by the loudspeaker into the printed vocal tract. Table 6.2 shows the measured formant frequencies. Another small error is observed when varying gain slightly, with a maximum MAE over this range of 1.02 %. Notably, there is no quantifiable scale for these changes

in gain on the Vocal Tract Organ. These variations are harder to explain, however some insight is presented by the Praat user interface. The Praat spectrogram viewer prints the fundamental frequency, which should be the same as the input frequency here, onto the display. For the gain variation measurements, Praat shows that the fundamental frequency is changing by around 4 Hz over this range of gain values. This likely suggests that there is some non-linearity in the speaker and that the output frequency is dependant on the gain. A fixed gain was used to avoid this.

Table 6.2: Formant frequencies, and MAE, measured at the lips of a 3D printed /ɜː/ tract when varying the gain of the Vocal tract Organ.

Vocal Tract Organ Gain	Formant Frequency (Hz)					MAE
	F1	F2	F3	F4	F5	
Control Gain	708	1318	2942	3951	5441	-
Decreased Gain	709	1331	2935	3895	5314	**1.02**
Increased Gain	715	1323	2951	3955	5430	**0.40**

With the characterisation measurements complete, the internal measurements were performed. From the recordings made at each measurement site, a set of extracted formant frequencies were calculated using the Praat LPC algorithm. Extracted formant frequencies were averaged over recordings approximately 3 s to 5 s in length. Errors on each extracted formant for each node, the MAE across each node, and the average errors across each formant can be seen in Table 6.3. MAE at the lips of the model shows agreement with the previous data from Table 5.1. The errors at each of the nodes infers much more complicated behaviour, however.

Table 6.3: Percentage error on individual extracted formants and Mean Absolute Error (%) (MAE) across all extracted formants, and excluding the second formant, comparing simulated acoustic propagation and physical internal measurements on a 3D printed /ɜː/ vowel vocal tract. Nodes represent measurement locations which are spread equidistantly throughout the tract from the glottis to the lips (The glottis, or node 0, is not included here). Blank cells appear when the Praat LPC algorithm found only four formants for a given output.

Output Location	Percentage Formant Error					MAE	MAE (No F2)
	F1	F2	F3	F4	F5		
Node 1	-31.93	-40.04	4.38	6.10	-	**20.61**	**14.14**
Node 2	-35.43	-34.97	-14.99	1.60	-8.10	**19.02**	**15.03**
Node 3	-29.33	-25.24	-4.23	-1.07	-13.71	**14.72**	**12.08**
Node 4	-20.99	-50.18	-15.37	3.70	-7.64	**19.58**	**11.93**
Node 5	-31.52	-56.78	-11.10	-13.33	-6.72	**23.89**	**15.67**
Node 6	-17.90	-46.48	10.81	0.70	-	**18.97**	**9.80**
Node 7	-11.30	-45.71	-10.46	-28.52	-9.78	**21.15**	**15.02**
Node 8	-2.23	-45.40	-6.54	-4.63	-17.56	**15.27**	**7.74**
At Lips	1.79	-36.88	2.03	-2.17	-10.09	**10.59**	**4.76**
Formant MAE	20.27	42.41	8.88	6.87	10.51	**17.79**	**11.63**

Average MAE across all nodes is 17.79 %, which is 7.2 % higher than errors observed of output at the lips. The standard deviation of the MAEs across all output locations is 3.78. Error is in fact minimised at the lips of the model, with every other node having a higher MAE. It is somewhat surprising that the acoustic simulation produces outputs which seem to only be accurate for the usually desired output of speech sound as it exits the tract, but there are several potential explanations.

As has been discussed in various prior sections, the simulation accuracy, previously only concerning output at the lips, is highly dependent on the geometry of the tract, the resolution of the tract, and the acoustic admittance. The quantitative

studies performed as part of Section 4.4.4 attempted to optimise these parameters for the similarity of the simulation output at the lips to physical measurement of 3D printed vocal tracts also measured at the lips. What seems likely is that these values do not maximise accuracy within the entire domain as previously assumed, but instead are particularly well suited to produce good outputs at the lips. One would assume that, as more optimisation was applied during this research, the simulation routine would become increasingly well-fitted to this lips output and may have drifted away from parameters that produce overall accuracy throughout the domain. Normally this would not be a concern as research into speech synthesis does not usually consider acoustic propagation within the tract as long as the output is good, however here it does impose limitations on the simulation's use when considering the way the acoustic field develops.

Errors may instead be more closely tied to the major difference between the physical measurements and the simulation: the domain boundaries. As previously discussed, and shown in Figure 4.22, reflections off of the domain boundaries do have a noticeable, but as yet unquantified, effect on the output of the simulation. The optimisation process, which includes the admittance at the domain boundary, is likely to have incorporated these reflections as though they are a physical behaviour and used them as a way to tune the simulation output closer to the recorded sounds. The physical measurements obviously do not have these domain boundaries. Furthermore, these internal measurements were performed in a different room to the measurements used during the optimisation process. As the domain boundary reflections also help to model the effect of the room beyond the tract in the physical measurements, it is possible that the values chosen were more specific to the environment in which the physical measurements were made than previously assumed.

A closer inspection of the errors on each individual formant reveals another interesting trend: the simulation was particularly poor at reproducing the second formant peak. When comparing to the averages of the VTSim measurements in Table 5.1, we can see that the previous set of measurements produced F4 and F5 with

good accuracy, F1 and F2 with reasonable accuracy, and F3 with low accuracy. Here F3, F4, and F5 were reproduced quite well, F1 was reproduced with low accuracy, and F2 was reproduced with very poor accuracy. When considering just the output at the lips this is even more apparent, with very good accuracy for all formants except F2. Table 6.3 also contains MAE values for each node when disregarding any values for F2. Average MAE is reduced by 6.16 % with most nodes being within approximately 15 % which is in line with the prior works by Gully et. al [4, 112]. Standard deviation of the MAEs across all output locations decreases to 3.52. This is only a small change from the original standard deviation, and suggests that values for F2 are consistently extracted from the data at a frequency which is inaccurate rather than the formant calculation algorithm failing to consistently locate the second formant.

While simply ignoring a formant is not acceptable scientifically, it does show that much of the discrepancy may be directly tied to the frequency band which the second formant occupies. F2 frequencies for the /ɜː/ vowel model in the internal measurements lie in a range of 877 Hz to 1472 Hz and in the simulated /ɜː/ vowel model they range from 1953 Hz to 2055 Hz. It is possible that some part of the simulation routine causes frequencies around 1 kHz to be shifted up to 2 kHz, but this is hard to test. A close inspection of the actual formant frequencies also reveals that the variation in formant frequency along the nodes is much greater in the physical measurements. The average range of formant frequency across all nodes is 229.4 Hz in the simulated data and 728.8 Hz in the physical measurements. This difference in range is most likely due to some combination of effects between the inaccuracies of the simulation at domain boundaries and of the effects of the wider measurement environment. A further study could be done in a future work which investigates the effect of the physical measurement environment by performing measurements in acoustically different environments, however this is beyond the scope of this work. Whatever the source of this error is, it is still only having a particularly large effect on the frequency region which F2 occupies and may not completely invalidate the

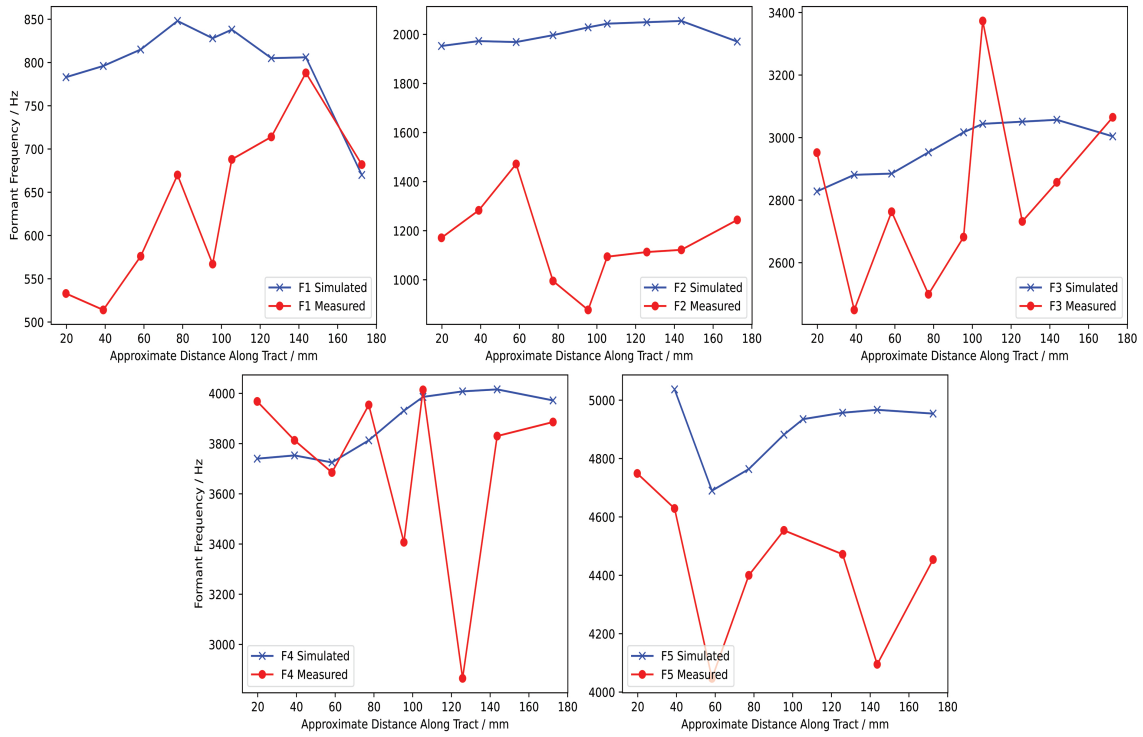


Figure 6.4: Frequency variation of each extracted formant frequency along the length of the tract from the glottis to the lips. Data in red was measured physically using a real 3D printed vocal tract model and data in blue was simulated from the same model.

results.

While using this simulation method to make choices and observations referring directly to formant frequencies may be difficult, it is possible that it can still be used in applications that are more concerned with the behaviour of formant frequencies rather than their exact values. For example, if a person is interested in increasing the frequency of their F3 for a particular reason, this simulation may still be able to provide particular areas of the tract whose expansion or contraction effect F3 the most. Figure 6.4 shows frequency variation for each extracted formant along the length of the tract, comparing simulated and physically measured extracted formant frequencies. Generally, both data sets look very different qualitatively. There are some shared trends and relationships in the data, but there is little overlap and the simulated data is much less varied than the physical measurements as previously discussed.

One final note to make is that it is possible that the physical measurements are

the ones which are not accurate. While great care was taken to ensure the external factors that may affect the measurements were minimised, there are many internal factors which could cause issues. In the simulation the vocal tract is modelled inside a cube which is large enough to contain the entire tract, with some additional propagation space, which is given the physical parameters of the walls of the tract from prior research of real vocal tracts. The physical measurements in this work only contain the tract within a 2 mm thick wall made of PLA. This set-up was previously tuned for the qualitative similarity to the original recordings as mentioned in Section 6.1, but may not be valid when considering internal propagation. A lack of reflections of acoustic pressure back in to the tract from interfaces between the airway and the outside of the skin may lead to unphysical behaviour.

6.2.1 Re-simulation With Identical Geometry

An additional simulation was performed which focussed on aligning the simulated geometry to the physical geometry of the 3D printed vocal tract as closely as possible. In contrast, simulations so far have always focused on aligning the simulated geometry to that of a real living vocal tract. Instead of encasing the vocal tract airway in a solid cube with the material properties of flesh, the tract was simulated with only a 2 mm thick wall in an otherwise empty space. The potential differences in outputs caused by this change has been neglected until this point in this research. The outer surface of the vocal tract does present a second surface over which voxelisation artefacts may appear, likely doubling the error in volume discussed previously. Table 6.4 shows the results of this re-simulation, both in full and excluding F2. An increase in accuracy is seen in the reproduction of F1, F2, and F3 with slight losses in F4 and F5. The average of the MAEs for each formant has also improved by 2.57 % down to 15.22 % which now brings the average accuracy in line with prior works by Gully et. al [4, 112]. A decrease in the standard deviation of the MAEs across all output location is also observed, from 3.78 down to 2.12 showing significantly less variance in the data set.

Table 6.4: Percentage error on individual extracted formants and Mean Absolute Error (%) (MAE) across all formants, and excluding the second formant, comparing simulated acoustic propagation and physical internal measurements on a 3D printed /ɜː/ vowel vocal tract. Nodes represent measurement locations which are spread equidistantly throughout the tract from the glottis to the lips (The glottis, or node 0, is not included here). Blank cells appear when the Praat LPC algorithm found only four formants for a given output. This simulation data was produced with geometry matching the physical measurements as closely as possible.

Output Location	Percentage Formant Error (Hz)					MAE	MAE (No F2)
	F1	F2	F3	F4	F5		
Node 1	-27.38	-40.35	1.62	-1.76	-5.19	**15.26**	**8.99**
Node 2	-20.80	-31.21	-13.92	-4.84	-7.84	**15.72**	**11.85**
Node 3	-10.28	-19.65	-1.64	-7.16	-18.67	**11.48**	**9.44**
Node 4	3.72	-46.22	-11.60	-0.40	-11.50	**14.69**	**6.81**
Node 5	-10.85	-51.22	-2.90	-12.86	-7.50	**17.07**	**8.53**
Node 6	5.85	-41.37	19.06	2.32	-	**17.15**	**9.08**
Node 7	14.06	-35.18	4.12	-24.43	-9.89	**17.53**	**13.12**
Node 8	25.48	-35.33	7.45	-0.60	-17.21	**17.21**	**12.68**
At Lips	6.73	-31.98	6.17	-5.08	-10.69	**12.13**	**7.17**
Formant MAE	13.90	36.95	7.61	6.61	11.06	**15.22**	**9.79**

Of note is the 1.54 % loss in accuracy at the lips of the vocal tract model. This corroborates some of the hypotheses presented in Section 6.2. The simulation routine and its parameters are likely tuned for accuracy at the lips while the tract is inside a solid cube, which is the scenario that all optimisation of the routine was done for. Changing the simulation environment has improved the physical accuracy of the simulation around the internal measurement nodes but has shifted the physical environment at the lips away from that for which it was optimised. It is likely that a second round of optimisation would drastically improve the results presented in Table 6.4. It is difficult to justify that round of optimisation however, as the

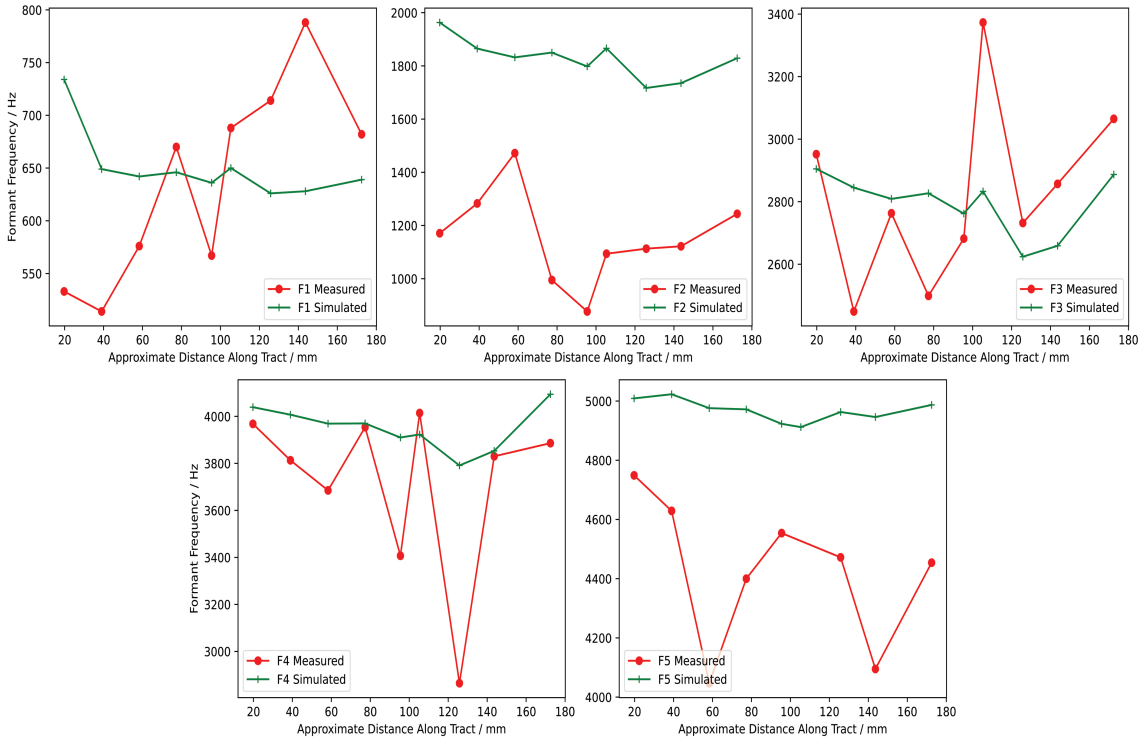


Figure 6.5: Frequency variation of each extracted formant frequency along the length of the tract from the glottis to the lips. Data in red was measured from a real 3D printed vocal tract model and data in green was simulated from the same model. Simulated data here differs from that in Figure 6.4 in that the geometry simulated was designed to match the physical measurements as closely as possible.

likely use cases for this research are much more closely related to the geometry of the vocal tract in a real human subject than they are to the exposed tract used for these measurements. Ideally the physical internal measurements could be performed on a vocal tract which was contained within a cube of material, however this would create issues with placing the microphones for measurement and with matching the material properties of the simulation and the measurements.

Figure 6.5 shows the frequency variation in each extracted formant along the length of the tract for the newly simulated data. When comparing to the data shown in Figure 6.4, the relationship between extracted frequency, which again likely more closely links to the varying amplitude of frequency components, and distance along the tract in the newly simulated data is very different. While neither data set perfectly match the relationship shown in the measured data, the newly simulated data is a closer match than the previous data set based on the MAE data

presented in Table 6.4.

A thorough optimisation of this simulation routine to maximise accuracy of these internal measurements on 3D printed vocal tracts which are as accurate as possible to real human vocal tracts is possible, but beyond the scope and timeline of this research. To better match realistic geometry, the 3D printed tract would need to be within a solid volume which would match at least the external shape of the subject's head, and would need to be printed in a material which has similar acoustic properties to the ones measured from real human vocal tracts. In addition, special care would need to be taken in providing holes for the microphones to be lowered into that do not drastically change the acoustic properties of the surround area. This is left as a future extension to this work but would likely lead to a significant improvement in accuracy of formants produced along the tract. It is difficult to ascertain what level of accuracy would be required for this simulation in order for it to be used to confidently inform any high stakes applications such as medical procedures, but it is likely significantly beyond that which this work has shown so far. The level of accuracy shown within is much more applicable to teaching and instruction uses though, and could likely be used in those settings to great effect.

As an example, the region around 110 mm along the tract in these models is at the back of the mouth. Figure 6.5 shows that the extracted F3 increases at this point in both the simulated and measured data sets. Based on arguments presented in Section 5.2.2, this implies that the space at the back of the mouth has geometry that has a resonance near F3 but at a higher frequency, thus there will be an amplitude peak in the power spectral envelope that leads to a raised extracted formant frequency. The prominence of this peak implies that it has a large impact on the actual formant value of F3. As such, if a a speaker is attempting to lower the frequency of their F3 formant for a particular reason, an effective method may be to attempt to lessen the effect of this resonance at the back of the mouth by reducing the constriction there or "opening up" that part of their mouth. Similar statements could be made at any point in the data that a large peak or trough appears, which

would become more prominent with more optimisation.

6.3 Chapter Summary

The assertion of accuracy in the model based on accuracy at the lips was tested in this chapter. These tests were performed using a novel measurement technique which allows for direct probing of the acoustic field at various points within a physical vocal tract model. Tests show some agreement between simulation and measurements, but accuracy for internal outputs is approximately 5 % lower than stated previously, on average. This level of accuracy was reached only after performing some re-alignment between the simulation domain and the physical measurements, which suggests that misalignment between the two is the source of other inaccuracy seen throughout this work so far.

A future work could perform a large breadth of analysis and optimisation to improve accuracy of internal vocal tract outputs by producing physical vocal tract models which are closely aligned to the condition of a living vocal tract in-situ. However, the measurements and simulations can still provide useful insight into the links between vocal tract geometry and the evolution of the acoustic field. These links can be used to suggest actionable changes for a hypothetical subject in speech training or therapy.

Chapter 7

Visualisation of the Vocal Tract

As touched upon in Section 4.1, a great deal of the work presented in this thesis has been undertaken with its eventual presentation and visualisation in mind. The early work on presenting the human vocal tract in a way which is approachable and intuitive to non-technical audiences can be seen in Section 4.1.2 in Figures 4.6 and 4.7. Both works involved presentation of the vocal tract of Nesyamun 'True of Voice', with the former through the medium of Lego and the latter presenting the tract as a virtual fly through alongside a living subject's tract within the video game Minecraft with a simple audio playback of the output of the physical vocal tract model. As proofs of concept, these two visualisations presented a subject which was previously only accessible by those with a high level of technical knowledge within mediums which are accessible by hundreds of millions of people of all ages.

While these visualisations help to give an initial introduction into what the human vocal tract really looks like, and can be used as a 3D diagram of the different parts of the anatomy of the tract, they don't provide much if any insight into the way that the shape of the tract gives rise to the speech sounds that we hear every day. Accomplishing this task requires implementing the way the sound is produced in the tract into the visualisation itself. A visualisation which makes some of the mechanisms of speech production intuitively accessible would be achieved by combining audio and visual components.

7.1 Vocal Tract Fly-Through

A virtual, 3D, fly through of a vocal tract corresponding to the English phoneme /ɜː/ was presented alongside a poster titled '**Virtual Exploration of The Human Vocal Tract**' as part of the Voice Foundation Symposium in 2021 [1]. This visualisation had a number of goals based on the progress of this research at that time. The first of these goals was to provide a scaled-up model of a vocal tract which a user could move around within. A 3D visual aid such as this could be useful as a companion to the 2D diagrams often used to explain the anatomy of the vocal tract. These 2D diagrams lack information on the relative size and shape of features and how much volume they occupy. It is also generally difficult to assess how the different features are connected in 3D space. All of these limitations of 2D diagrams are solved when pairing them with a 3D version of those same diagrams. The second goal of this visualisation was to provide audio output alongside the 3D view to begin to develop a visible relationship between the geometry of the tract and the way the acoustic field evolves as it travels along it. This would be done by playing back simulated audio within the tract at regular intervals and within certain side passages. Finally, it was considered desirable to present this visualisation within an immersive 3D medium such as Virtual Reality (VR) to better convey relative sizes and shapes through the use of stereoscopic displays.

These goals required the use of a modern 3D rendering engine with support for user interactivity and VR. In Section 3.1, the Unity game engine was used to test the Google Resonance Audio plug-in as a potential simulation method for this work. The first task in performing this test was importing the vocal tract geometry into the rendering engine. At the time of creation it was decided that importing the STL files directly into the engine, rather than converting the files into a format that was supported by Unity, would minimise any variations between the visual geometry and that which had been simulated beforehand. The Parabox STL plug-in for Unity was used as the STL importer for this visualisation [122]. This plug-in deals with importing of the meshes from the binary STL file, splitting those meshes into

multiple parts to avoid Unity's maximum vertex counts on imported assets, and then re-merging those meshes into one asset whilst preserving relative transformations. Parabox also supports ASCII STL files however, as the files used in this research are typically optimised for 3D printing, all the STL files used are already in the binary format.

With the geometry imported, some logic must be written to allow the user to explore that geometry. Unity is most commonly used for games development and so has a wide variety of pre-built 'character controllers' which can be used to accept inputs from the user and translate those inputs into movement in the space. Most of these character controllers are designed for walking around on surfaces in 2D or 3D space however, which is not applicable to this project. Instead, a custom controller was written that mimics the controls used to move the camera in 3D modelling software and in the Unity editor itself. Unity uses C# as its scripting language, so a simple script was written and attached to the player camera that allowed the user to move forwards and backwards relative to their perspective with the arrow keys, and rotate the viewport with the mouse while holding the right mouse button. This implementation proved to be an effective proof of concept, making intuitive sense and allowing for relatively free movement around the tract, but did reveal some design limitations that should influence future visualisations.

Even with a vocal tract that is increased in scale by a large factor, it can become disorienting to move around in tightly constricted spaces within the tract. This is emphasized when the user has spent enough time in the visualisation that they are no longer well oriented with the spatial axes of the scene. While a reset button that would realign the user's rotation with the scene would help with this, there are two underlying issues that should be solved if producing a new visualisation upon the basis of this one. The user's camera is currently not bounded by physical collisions with the tract walls. This means that a user is never restricted in their movement when close to the walls of the tract, but can lead to disorientation when a user accidentally moves through the walls to the outside of the tract. This is made worse

by the wall thickness of the model. Unity will render both surfaces of the mesh at all times, which means that a user can end up inside the walls of the model and not sure how to get back to the inside of the tract. A solution to this issue would likely be removing the wall thickness so that there was only one surface to render, and adding a small occlusion zone around the user within which the walls of the tract become transparent. This would ensure that the user never gets too close to the walls and loses track of their location, and would also make it easier to move in and out of the tract purposefully. In addition to issues of physical restriction of the camera, repositioning the camera in 3D space without moving forwards or backwards is impossible and can make precise navigation difficult if not disorienting. To solve this issue, the ability to move left, right, up, and down should be added for finer control. The same could be added for rotational controls as well at the risk of increasing the complexity for the user.

With an enlarged vocal tract which the user can navigate around, the final step of this visualisation was to implement audio feedback based on the user's location in the tract. A series of small spherical objects were placed throughout the tract to act as speakers. A sound file was attached to each of these speakers and the radius of these sound sources was set to prevent a user from hearing two of the sound sources at once. Sound files were produced from acoustic simulation data as described in Section 4.3. As such this data was significantly less accurate than data produced by VTSim in Chapter 5 and lacks any of the physical validation as discussed in Chapter 6. Sound files used in this visualisation were also produced directly from using the source filter method with the calculated Vocal Tract Transfer Function (VTTF) and a Liljencrants-Fant (LF) model pulse. This method produces buzzy and unnatural speech sounds when compared to the Klatt synthesiser that was later implemented, but variation is still audible between the different sound sources as a proof of concept. Notably, the internal sound output was disabled in the version which was made available during the symposium due to concerns about the accuracy of the sound files.

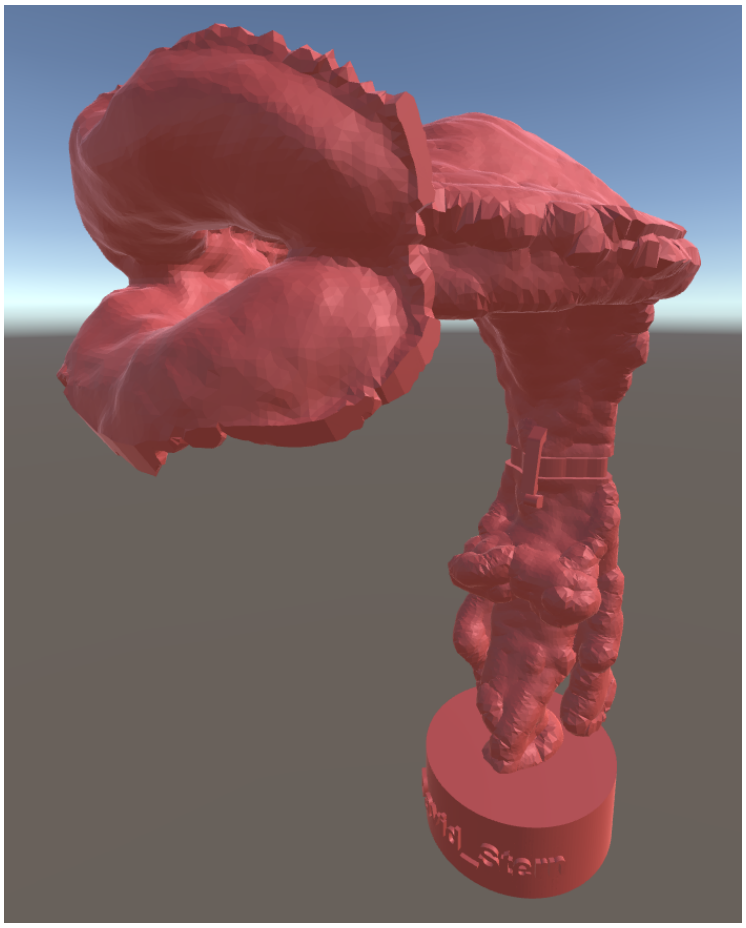


Figure 7.1: Intro scene for the vocal tract virtual fly through demo created for the Voice Foundation Symposium 2021. The user can move around the environment using the arrow keys and can pivot their view using the mouse.

Figures 7.1, 7.2, 7.3, and 7.4 show screenshots of this visualisation. They show a high resolution interactively explorable scene, with several parts of the anatomy labelled. By being able to move through the tract as they choose, users can get a stronger sense of how the parts of the tract are connected to each other and also of the relative size and constriction throughout the tract.

A visualisation such as this one could be useful to a variety of professionals in the field of speech. Phoneticians and linguists could use a visualisation such as this as a three dimensional diagram of the anatomy of the tract from which to teach the concepts of phoneme production based on vocal tract articulation. Speech therapists could use an interactive fly-through of the tract to show a patient which parts of the tract are responsible for different parts of speech production and what changing the shape of them might do to your speech characteristics. Ear, Nose, and Throat

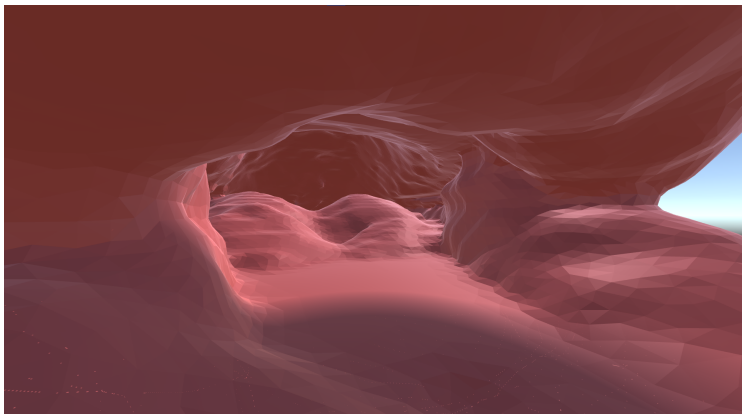


Figure 7.2: View into the mouth through the lips from the vocal tract virtual fly through demo.

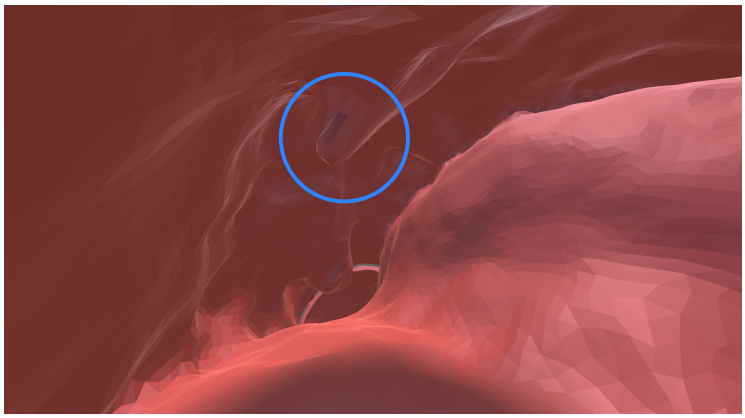


Figure 7.3: View down the throat from the vocal tract virtual fly through demo. The Uvula is circled in blue.

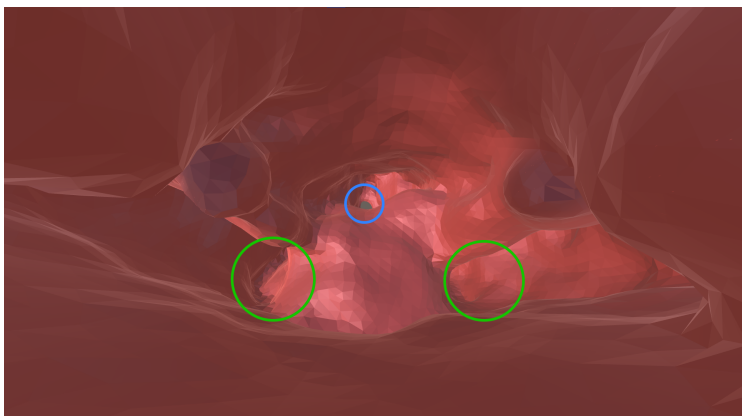


Figure 7.4: View towards the glottis from the vocal tract virtual fly through demo. The Glottis opening is circled in blue and the passageways leading to the Piriform Fossae are circled in green.

(ENT) surgeons could use this as a tool to explain how their tract will change and how that change controls the production of sound to people who are soon to undergo reconstructive surgery. Many other potential use cases likely exist.

7.2 Auralisation of a Vocal Tract

An interactive demo featuring a vocal tract corresponding to the English phoneme /ɜː/ was presented alongside a talk and poster titled '**Using Voice Synthesis Techniques to Virtually Explore the Sound Field within the Human Vocal Tract**' for the Voice Foundation Symposium in 2022 [2]. The goals of this visualisation were to act on some lessons learned in the creation of the demo described in Section 7.1, and also to implement the advancements in simulation accuracy produced by performing studies on the Arai vocal tract models. At the time of the production of this visualisation, the studies performed on the Arai models were still undergoing up to and including most of Section 4.4.1. While this simulation routine still contained some unexplored errors, it was significantly more accurate than the data used in the making of the visualisation presented in Section 7.1, and importantly now used a four formant synthesiser written in Pure Data for the production of speech sound [123].

Pure Data is an open source visual programming language designed for use in audio engineering applications. Pure Data has a wide array of built-in functions and tools for creating and manipulating audio signals, only a small subset of which were used in this work. Figure 7.5 shows the front panel that was used to synthesise the speech sounds for this visualisation and Figure 7.6 shows the logic that powers that synthesis. Each of the numbered buttons will input the first three formant frequencies that were calculated in Praat for simulation outputs at successive nodes through the vocal tract, with '1' being closest to the glottis and '11' being just outside the lips. These three formant frequencies are used as bandpass filters that a source signal is passed through separately before being mixed back together into a single sound signal which is then played using a digital to analogue converter.

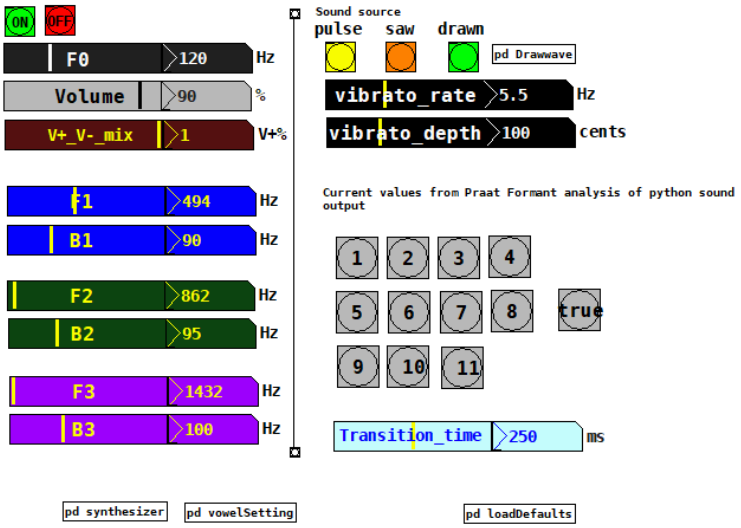


Figure 7.5: The front panel of the speech synthesiser used to create the sound files for the auralisation of a vocal tract. This Pure Data patch allows for manual setting of the formant frequencies and bandwidths for the first three formants or quick switching between pre-defined values with the buttons on the right of the patch.

The top half of the 'graph' shown in Figure 7.6 uses the vibrato rate and depth set on the front panel, as well as an excitation waveform to create a voiced source signal with fundamental frequency also set on that panel. This voiced source is then optionally mixed with a voiceless source signal to make a resultant vocal source for the synthesis. The bottom half of the graph creates a bandpass filter using each formant's frequency and bandwidth and then applies that filter to the vocal source, then mixes the three filtered signals back together.

This synthesiser 'patch', the name of a program in Pure Data, was a resource already available within this research group and has some default settings that were left as given. Only the formant frequencies and bandwidths were altered for this research. Notably, this synthesiser only uses the first three formants which is in contrast to the rest of this work which has always stated at least four formants if not five. As this patch is intended for creating natural speech sounds, it is only concerned with the first three formants which are generally considered to be all that is needed to have good intelligibility and enough variation between different speech sounds to be able to differentiate them. The work presented here so far has typically had good success with recreating the first, fourth, and fifth formant

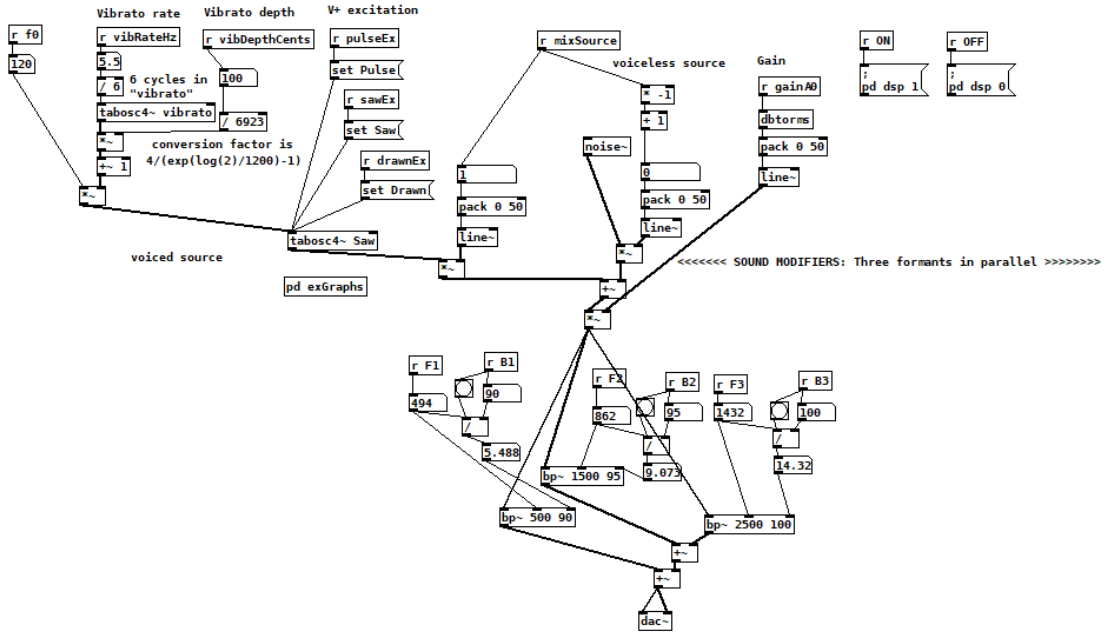


Figure 7.6: The logic which performs the speech synthesis used to create the sound files for the auralisation of a vocal tract. Solid lines represent the flow of data between control nodes which perform functions on the incoming data.

and poor accuracy on the second or third formant depending on what stage of this research is considered. Using a three formant synthesis does require the high error formant to be used in the synthesis, and means that the error of that formant will have a larger effect on this synthesis based on a smaller number of formants. Despite this the acoustic data used in this visualisation is still a large increase in accuracy when compared to the previous visualisation, and the use of a more sophisticated synthesiser greatly improves the intelligibility of the sound files which makes the user's comparison between them during the visualisation much easier.

The visualisation was made using the Unity game engine for the reasons already stated. The view presented to the user can be seen in Figure 7.7. Instead of importing a full vocal tract into this model and allowing for a full 3D exploration of the tract, it was decided that this simulation would function best in 2D. This avoids the issues discussed in Section 7.1 with navigation around the 3D space, and also keeps the focus of the visualisation on the way the sound changes throughout the tract rather than the physical exploration of its geometry. A 2D viewpoint required bisection of the tract, which was performed in Blender, to allow the user to see into

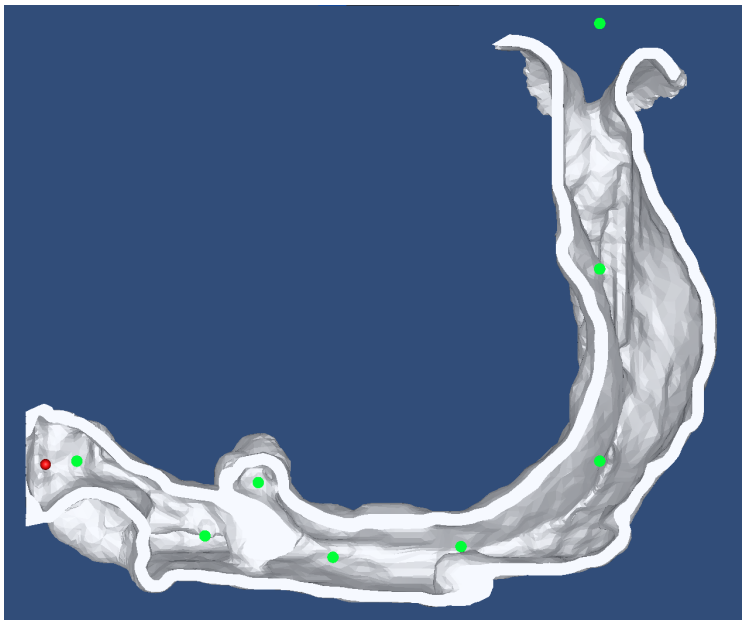


Figure 7.7: Vocal tract internal sound profiling demo created for the Voice Foundation Symposium 2022. The red sphere is the location that the user is listening from and can be moved around the space using the arrow keys. Each of the green circles is a virtual speaker that plays the synthesised sound at that point, produced using acoustic simulation data.

the airway of the tract. Along the airway, a number of green discs were placed that would act as the speaker locations for the various sound files. Again, the radii of these sound sources were set so that the user would only hear one source at a time. A unique feature of this visualisation when compared to the previous one and the work done after this visualisation was produced, is the presence of a sound source in the valleculae. This was one of the first outputs of this work that attempted to show the sound in the tract at a point which was separated from the main airway and in an area which is geometrically different from the rest of the tract.

While this visualisation is significantly easier to navigate and presents the acoustic data in a much more intuitive way, the 2D view does still obscure a lot of the geometrical variation that is present in the tract. For example the existence of the piriform fossae is not particularly clear, only one of the vallecula can be seen, and the way in which the valleculae are connected both to themselves and to the main part of the airway is not clear. A future visualisation would likely combine the high accuracy and natural sounding synthesised outputs using the method from this vi-

sualisation and the data from Chapter 6, with a 3D fly-through like the one shown in Section 7.1.

7.3 Chapter Summary

Two visualisations created during this work for the purpose of presenting the vocal tract both visually and aurally in an interactive and intuitive manner are described in this chapter. In the first visualisation, a 3D fly through of the tract is presented that allows a user to move through the vocal tract in 3D to get a hands-on idea of how the different parts of the tract are connected. This fly through was presented alongside some primitive synthesised sounds produced along the tract as a proof-of-concept for a piece of interactive software that allows a user to directly relate physical geometry to audio output.

This proof-of-concept was built upon in the second visualisation, which included more sophisticated synthesised speech output in a simpler to navigate albeit less informative 2D space. By moving their microphone around the bisected vocal tract and between the virtual speakers, users could directly compare and contrast synthesised speech sounds at different points in the tract easily. The lessons learned in creating these two visualisations pave the way for a combined one which would present multiple tracts in an interactive 3D space with sound outputs within them and at their lips, possibly incorporating vocal tract geometry manipulation in real time.

Chapter 8

Conclusions and Future Work

This thesis has presented a novel technique for simulating the acoustic propagation through a human vocal tract, verified that simulation through physical measurements, and presented both the simulation routine and data it has produced in an accessible manner. In this chapter, a summary of the thesis and its contributions is provided by chapter.

8.1 Thesis Summary

Chapter 2 provides an introduction to the physical processes of acoustic propagation in the human vocal tract and how those processes lead to speech. A way in which those processes can be used to synthesise artificial speech sounds is also given. The introduction provided in this chapter is brief but is expanded upon throughout this thesis and provides a foundation from which to consider the simulation of the human voice. A wide array of acoustic modelling procedures are also presented, including both geometrical and wave-based methods. While geometrical methods are much easier to implement and can lead to significant reductions in simulation time, the accuracy provided by wave-based methods to the underlying physical processes can not be overlooked. Wave-based methods are the obvious choice when requiring high accuracy such as in speech science. Each of the wave-based methods discussed is presented fully mathematically or in terms of its relationship to a previ-

ously described method. This chapter also includes a brief discussion of some of the modern advances in acoustic simulation, including the use of high performance multithreaded hardware such as GPUs and some methods used to maximise immersion in VR applications.

A series of pre-existing modelling packages are introduced in **Chapter 3**. Each investigated modelling package is briefly explored and then discussed with respect to its potential advantages and disadvantages in vocal tract acoustic modelling. While many of these modelling packages are mature and widely used, they all have certain drawbacks which limit their applications to this work, for example difficulty of defining 3D geometry, incorrect handling of sufficiently small features, and relative inaccessibility due to licensing costs and long simulation times.

Chapter 4 shows the beginning of the development of an acoustic propagation algorithm designed for human vocal tract modelling. The main aims of this development separate it from prior research: a requirement for the entire propagation domain to be calculated and stored for the entire duration of the simulation for further analysis, and a requirement for the simulation routine and the data it provides to be accessible to both the wider scientific community and also individuals with less technical knowledge and less resources. An approach to 3D geometry which is not based on common engineering processes like Computer Aided Design (CAD) has led to the use of powerful 3D modelling applications such as Blender which provide a great deal of control of model preparation and editing. A forward eye to the eventual visualisation of simulation data also produced voxelisation as both a powerful visualisation tool and an extremely convenient one for importing complex general geometries into simulation routines.

Both the FDTD and DWM methods were explored in this chapter. Several issues were observed with the implementation and use of FDTD which largely arose due to the high complexity and frequent constrictions of the vocal tract geometry. As such this work made great use of DWM and its acoustic admittance term which implicitly allows for acoustically accurate propagation within the walls of the tract. Initial

simulations of acoustic propagation within sample tract models produced from MRI data were performed. Formant frequencies produced were frequently off from given values by a scaling factor between 2 and 5, possibly due limitations of the simulation routine, errors in the FFT algorithm used, or due to obtaining an arbitrarily scaled eigenvector of the system. Accuracy comparisons between this simulation routine and prior research showed similar results but required manual removal of the scaling factor and were based on given formant frequencies for various vowels rather than physical measurements. A set of simplified on-axis vocal tract models for Japanese vowels were then used to optimise the simulation routine for a variety of parameters using physical recordings of the real models. This allowed for direct validation of the simulation results and this technique is one of the novelties of this research. These optimisations were then applied to real physical tracts as a starting point for further optimisation in order to re-align the simulation routine to its targeted domain. This concluded with an average MAE of 12.77 % across three vocal tract models, which is in line with prior research.

Having produced and optimised the acoustic simulation routine, work was done to consolidate the entire process, from importing the STL file all the way to formant calculation and speech synthesis, into one acoustic simulation package dubbed 'VT-Sim' in Chapter 5. An in-depth description of the additional development required for this is given, and a series of accuracy comparisons made on the outputs of the combined simulation routine based again on physical measurements of 3D printed vocal tracts. Average MAE across the four vocal tracts simulated here is 10.12 % with a standard deviation across the MAEs of each model's formants of 2.691, with accuracy gains and variance reduction likely arising from testing on one additional model and minor improvements to code in the process of creating the package. A set of simplified uniform tubes were then simulated using VTSim to test its ability to recreate formants in a mathematically calculable example. The average MAE across both simulations was 7.435 with a standard deviation of 4.175. This shows good agreement, despite mismatches between the simulation domain and the ex-

ample acoustic environment. VTSim was then used to interrogate the evolution of the acoustic field along the length of the tract in each of the vocal tract models previously simulated. An explanation is provided for the way the extracted formant frequencies shift in relation to the geometry changes along the length of the tract using a simple analogy of resonance in a tube, showing the potential power of this method in providing a technical basis for intuitive explanations of how changes in the tract geometry can affect the voice. This method was then applied to a model which has had its piriform fossae removed to show the affect that this change has on the average formant frequency along the length of the tract.

In **Chapter 6**, attempts to validate the internal acoustic outputs being used at the end of Chapter 5 using physical measurements. By using a miniature microphone and a custom-built 3D printed vocal tract with measurement ports along its length, a series of acoustic measurements were taken of the evolution of the acoustic field within a physical vocal tract. The procedure for these measurements is described in good detail and is one of this work's novel contributions to future research. These measurements showed that while the simulation does provide results which are within a range of relative accuracy, the optimisations made for VTSim to produce outputs which are accurate at the lips of these models may not be as accurate for reproduction of the internal field. A first pass at improving the accuracy of the geometry of the simulation to that of the physical measurements, without the adjustment of the parameters of the simulation itself, led to a 2.57 % gain in MAE to an average of 15.22 % with a standard deviation of the MAEs across all output locations of 2.12. With more optimisation for re-aligned simulation, it is expected that accuracy similar to that previously attained at the lips would be produced. Performing this optimisation in a way which produces a simulation which is accurate to a real human vocal tract rather than a 3D printed vocal tract is complex though, and beyond the scope of this work. This will be discussed further in 8.4.

Finally, **Chapter 7** presents a few visualisations created during this work which present both the vocal tract and the acoustic field produced by it in interactive

and intuitive ways. Visualisations have been created for both the 50th and 51st Voice Foundation Annual Symposium. The first work presented in this chapter is a high-resolution interactive vocal tract which has been greatly scaled up to allow a user to move around within it. This visualisation could be used as a teaching tool to give a better understanding of the way the different anatomical features of the vocal tract are laid out. The second demo features an interactive bisected vocal tract that a user can move a microphone within, and spread throughout this tract are a series of speakers that play the sound produced at that position in the tract based on simulation outputs. This allows a user to both see and hear how the vocal tract changes along its length and how that change produces the speech sound at the lips.

8.2 Novel Contributions

This work provides the following novel contributions to this field:

- **VTSim** The all-in-one acoustic simulation package presented in Chapter 5 is purpose built to produce speech profiling data and output sounds for arbitrary vocal tract models. This package is written in Python and is freely available in its entirety at the following address: https://github.com/dotdandotunderscore/VTSim. This should greatly lower the barrier of entry to speech science in this field.

- **Direct Comparison to Physical Measurements** There is only a small prior body of work which directly compares simulated speech outputs to corresponding recordings of physical vocal tracts. The methods described within provide a strong validation procedure for simulation routines and are used in this work to show a high level of accuracy in the simulation.

- **Internal Acoustic Measurements of the Vocal Tract** Chapter 6 described the process and outcomes of performing direct physical measurements

of the evolution of the acoustic field within a 3D printed vocal tract. The process for performing these measurements and the results they produced in this work are well described and can be used for a variety of validation measurements beyond this work.

- **Interactive 3D Visualisation of the Vocal Tract** The visualisations created as part of this work present the vocal tract and the mechanism of speech production in a variety of ways which are both interactive, and intuitive. This includes presentation within extremely widely used software, and in custom-built applications that present the audio outputs alongside the geometry being explored.

8.3 Hypothesis Revisited

The hypothesis presented at the start of this thesis is as follows:

By simulating the acoustic field throughout the entire vocal tract the evolution of speech sounds within the tract can be directly and quantitatively related to physical variations in the tract geometry.

In this thesis, a novel simulation packaged dubbed 'VTSim' has been produced which is capable of performing an acoustic simulation of a human vocal tract with arbitrary geometry that stores the pressure within the entire propagation domain at every time step. This model, through direct comparison to physical vocal tract measurements, has been shown to be capable of reproducing formants at the lips with an average accuracy of 10.12 % with a standard deviation across the MAEs of each model's formants of 2.691 with no additional treatment or consideration of the results. This same model has then been used to extract formant frequency measurements within the tract, which have also been directly compared to physical measurements performed using a bespoke measurement set-up as described in Chap-

ter 6. Average accuracy across nine measurement points for one vocal tract model was 15.22 % with a standard deviation of the MAEs across all output locations of 2.12 with minimal optimisation and no additional treatment of the results.

In this work the behaviour of acoustic outputs taken within and along the vocal tract have been compared to the geometry within which they are taken, and explained using relatively simple analogies and physics concepts. For example, Section 5.2.2 contains discussion of the frequency shifting of extracted formant frequencies in a vocal tract model when moving from the constricted area at the back of the throat to the more open area within the mouth. A simple application of basic acoustics corroborates the frequency changes which are seen in the simulated data set. Section 6.2.1 uses similar arguments to suggest that a change in geometry at a certain position in the tract would lead to a noticeable frequency shift in specific formants. While knowledge of how to articulate the throat to create different sounds has been developed over long time periods empirically in the practice of speech, the ability to see how those articulations affect the evolution of speech in the tract quantitatively before this research is relatively inaccessible. Being able to explore the tract quantitatively in this way will also likely lead to further discoveries and uses that were not possible based only on speech outputs of an individual.

For example, conventional wisdom in speech coaching and singing instruction may recommend the expansion of a certain part of the mouth or throat to produce a particular sound, which could also be produced by instead making other changes which may not have been immediately obvious. These kinds of suggestions are common in speech therapy that aims to assist a speaker with adjusting the way their voice sounds, often targetting individual formant frequencies directly. The granularity with which the acoustic field of the tract can be probed in this way also paves the way for great precision with instruction and even medical procedures. To continue the previous hypothetical, an instructor could not only instruct an individual to expand a certain part of their vocal tract to achieve a certain sound, but could directly quantify by how much the individual would need to do so. In the

field of medicine, doctors may be able to take MRI scans of a subject and compare the output of a simulation on those scans to historic recordings of that subject's voice to help plan surgery to recreate it or even to edit it. A process such as this could be used to provide targetted insights into the effect of physical vocal tract anatomy trauma on the spoken output of a speaker, and influence a plan of action to remedying the effects of that trauma on that subject's speech.

While the work presented within this thesis is promising, there are still issues with accuracy and implementation. While the 10 % average accuracy stated for the VTSim outputs at the lips in Table 5.1 is good in comparison to other works, the loss of accuracy seen in the internal measurements presents a barrier to the use of this research at this time. Given more time, a large amount of work would likely have been done in improving this accuracy until it approached the previous levels. Better still, an experiment designed to perform optimisations of the simulation routine on a set of physical models which are designed to match real vocal tracts in-situ as closely as possible would have been performed. Section 8.4 presents many areas in which this work could be improved upon and advanced in the future.

Within the scope of this work, this researcher believes that the goals set out in the hypothesis have been clearly met with many interesting avenues for expansion. With further research on top of the ground laid here, alongside more validation and optimisation of the simulation and comparison to more living subjects, a simulation method such as this could likely have a wide array of potential uses across the field of speech science.

8.4 Future Work

8.4.1 Improvement of the DWM Simulation Routine

While the DWM simulation method described in this work has been shown to be robust enough to reproduce formant frequencies with an average MAE of approximately 10 %, there are still many functionalities which are not present in this routine

that could provide meaningful gains in overall accuracy. The first obvious improvement to the algorithm is in the implementation of a frequency-dependent acoustic admittance parameter. Boundaries in real acoustic media reflect acoustic waves with a different phase and amplitude to the incoming wave in a way that is frequency dependent. Characterising and implementing some kind of function which computes the acoustic admittance in each waveguide during run time based on the incoming frequency may lead to a much more physically accurate simulation which would hopefully produce more accurate outputs. This is a non-trivial process however, and is not equivalent to the method of implementing frequency dependent boundaries in the FDTD scheme due to the lack of acoustically hard boundaries within the domain. This would also add significant simulation time increases if not intelligently implemented due to a necessity to compute the acoustic admittance at every time step.

The next weakness of this simulation is in the implementation of the acoustic source. In VTSim, the source is placed at a single node which is in the centre of the glottis on the bottom of the model. In a real vocal tract, the volume velocity which acts as the input to the vocal tract is produced by the combination of airflow from the lungs and the vibration of the glottis. As such, to accurately model this, the glottis should likely be modelled as a surface that produces the volume velocity source across itself evenly. In addition to this source directivity, which describes the amplitude spectrum of a source from its origin in 3D space, is not considered in this work [124]. In this simulation the source is entirely omnidirectional, which may also lead to a loss in accuracy.

An inaccuracy which is seemingly not considered in any research which has influenced this work is that the speed of sound in the acoustic propagation does not differ in the airway and in the wall of the vocal tract. The speed of sound is used to determine the sample frequency due to the Courant convergence condition, alongside the spatial resolution. To account for the speed of sound would either require updating the acoustic propagation within the airways less frequently than within

the walls or making the mesh grid that propagation is computed on significantly more sparse within the walls. Both of these options are incredibly non-trivial to implement, with special care taken to avoid feedback loops caused by acoustic pressure entering a wall, moving along the tract, and then re-entering the airway travelling in the opposite direction. This would likely require some complex improvements in directionality of propagation as described prior.

Finally, the lack of an absorbing boundary layer at the edges of the propagation domain leads to significant non-physical artefacts in this work that may lead to large improvements in accuracy if removed. In this work, the LRS formulation popular in FDTD schemes has been tuned to maximise accuracy however there is still significant reflection off of the boundaries and back into the domain. It is not possible to determine the effect of these reflections without first implementing a solution, but it is expected that the effects are profound especially for internal acoustic measurements. Much prior work has attempted to formulate a true absorbing boundary condition for the DWM but none as yet are as effective as the PML available in two-field applications [125, 126, 127].

8.4.2 Improvement and Validation of VTSim with Internal Acoustic Measurements

Chapter 6 presented a first attempt at characterising the acoustic field within the vocal tract and using measurements to validate the simulation outputs. While this validation did show some agreement, there is a clear loss in accuracy when compared to measurements just considering the output at the lips. This loss in accuracy can also be seen in the mismatch between the relationship between formant frequency and distance measured along the tract seen across the physical measurements and the simulations. Adjustment of the simulation geometry led to an immediate 2.57 % improvement which suggests that the simulation routine is incorrectly optimised for internal measurements, specifically of 3D printed tracts.

To adequately perform these optimisations in a way that matches the intended

use case, a physical vocal tract model which closely matches the acoustic properties of a real human vocal tract in-situ would be required. This includes: encasing the vocal tract in a material which is acoustic similar to flesh, mounting the tract to a loudspeaker in a way which does not adversely affect the acoustic properties around the mounting point and which does not introduce any inaccurate vibrational modes in the assembly, and designing the boreholes used for the physical measurements such that they can be plugged when not in use and do not affect the acoustic environment when they are in use. If all of these requirements can be achieved then this physical measurement set-up could be used as a calibration for VTSim in terms of varying model resolution, acoustic admittance, boundary admittance, and any other relevant physical parameters. Only by performing this calibration could the accuracy of the internal VTSim outputs be considered to have good accuracy and be used with confidence.

On the subject of validation of the simulation, a large study on recreating simple acoustic examples could also be performed to discern the simulation's accuracy at producing a known ground truth, and to better understand the effects of the formulation of the simulation domain on its ability to reproduce these results.

8.4.3 Comparison Between VTSim and Other Methods

While the Digital Waveguide Mesh (DWM) method used in this work for acoustic simulation is capable of accurately modelling the physics of acoustic propagation in the tract, there are many other methods of computing acoustic propagation which may have their own advantages and disadvantages. VTSim, instead of being used as a tool for producing data to be used in other applications, could be a useful tool for benchmarking and exploring state-of-the-art software and processes.

A great deal of processes for experimentally measuring vocal tract speech outputs have been developed and presented in this work, that could be used to validate any number of speech acoustics applications. For example, Section 3.4 described initial explorations of the multi-physics solver COMSOL which is a highly sophisticated

software package that could be used for the simulation of human speech sounds while accounting for complex interactions such as fluid dynamics. Comparing a COMSOL simulation of the vocal tract both to physical measurements on the same geometry and to VTSim would allow for a quantitative measure of the advantage gained by using this package, if any.

Performing a suite of investigations such as this would benefit from a deep knowledge base on whichever tool was being tested, and as such is beyond the scope of this work.

8.4.4 More Visualisation

Chapter 7 presented two visualisations that aimed to provide a way to explore the human vocal tract, both physically and aurally, in an interactive and intuitive way. Ideally, a third visualisation would have been made as part of this work that combines the advantages of both of these visualisations while solving some of their implementation issues.

This visualisation would likely be in full 3D, possibly within VR, with a well implemented way of moving around physically within the space. Multiple vocal tracts would be imported into the visualisation and each of these vocal tracts would have fully synthesised acoustic outputs throughout them. For the purpose of teaching and outreach, the tracts could be annotated with the anatomical features seen throughout, and signage that explains why the sound changes in the way it does would be found at key parts of the geometry.

This would be even more impactful if the user was able to perform slight changes to the geometry, either in preset ways or in real time, and hear the effects of those changes immediately. Implementing this in real time is a large computational challenge, but is not infeasible in the future.

8.4.5 Further Applications in Other Fields

As has been discussed throughout this work, the most interesting avenue of further work based upon this research is in applying it to other fields. Potential applications have been mentioned multiple times throughout, ranging from instruction in the use of the voice which is informed by simulation on an anatomical level, to the use of these simulation methods to determine the location and magnitude of surgical interventions with an aim of creating a specific voice in a subject.

The potential applications are likely even more broad than the simple ones presented here as a result of the particular background of this research but, with the availability of VTSim and the documentation present in this work, the discovery of those applications is left as an exercise to the reader.

Abbreviations

BEM Boundary Element Method

CAD Computer Aided Design

CUDA Compute Unified Device Architecture

DG Time-Explicit Discontinuous Galerkin

DWM Digital Waveguide Mesh

ENT Ear, Nose, and Throat

ESM Equivalent Source Method

FDTD Finite-Difference Time-Domain

FEM Finite Element Method

FFT Fast Fourier Transform

FVM Finite Volume Method

GPU Graphics Processing Unit

HRTF Head-Related Transfer Function

K-DWM Kirchoff Type-Digital Waveguide Mesh

LF Liljencrants-Fant

LPC Linear Predictive Coding

LRS Locally Reacting Surfaces

LTI Linear Time-Invariant

MAE Mean Absolute Error (%)

MRI Magnetic Resonance Imaging

PLA Polylactic Acid

PML Perfectly Matched Layer

PSTD Pseudospectral Time-Domain

SRL Standard Rectilinear

VR Virtual Reality

VTTF Vocal Tract Transfer Function

W-DWM Wave Based-Digital Waveguide Mesh

WAV Waveform Audio File Format

References

[1] D. R. M. Woods, "Virtual Exploration of The Human Vocal Tract," in *50th Anniversary Symposium - 2021 Symposium Program*. The Voice Foundation, 2021, p. 12. [Online]. Available: https://voicefoundation.org/wp-content/uploads/2021/05/2021-SYMP-Program.pdf

[2] D. R. M. Woods and D. M. Howard, "Using Voice Synthesis Techniques to Virtually Explore the Sound Field within the Human Vocal Tract," in *51st Annual Symposium - 2022 Symposium Program*. The Voice Foundation, 2022, p. 13. [Online]. Available: https://voicefoundation.org/wp-content/uploads/2022/04/2022-SYMP-Program_DraftFinal.pdf

[3] J. O. Smith III, *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects*, 1st ed. W3K Publishing, 2010.

[4] A. J. Gully, "Diphthong Synthesis Using the Dynamic 3D Digital Waveguide Mesh," Ph.D. dissertation, University of York, 2018.

[5] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West, "Beam tracing approach to acoustic modeling for interactive virtual environments," *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 21–32, 1998.

[6] Google, "Resonance Audio," 2018. [Online]. Available: https://resonance-audio.github.io/resonance-audio/

[7] E. Dick, "Introduction to finite volume methods in computational fluid dynamics," in *Computational Fluid Dynamics*, 3rd ed. Berlin Heidelberg: Springer-Verlag, 2009, vol. M, ch. 11, pp. 275–301.

[8] D. M. Howard, J. Schofield, J. Fletcher, K. Baxter, G. R. Iball, and S. A. Buckley, "Synthesis of a Vocal Sound from the 3,000 year old Mummy, Nesyamun 'True of Voice'," *Scientific Reports*, vol. 10, no. 1, pp. 2–7, 2020. [Online]. Available: http://dx.doi.org/10.1038/s41598-019-56316-y

[9] K. Kowalczyk and M. Van Walstijn, "Formulation of locally reacting surfaces in FDTD/K-DWM modelling of acoustic spaces," *Acta Acustica united with Acustica*, vol. 94, no. 6, pp. 891–906, 2008.

[10] G. E. Peterson and H. L. Barney, "Control Methods Used in a Study of the Vowels," *Journal of the Acoustical Society of America*, vol. 24, no. 2, pp. 175–184, 1952.

[11] T. Arai, "The Replication of Chiba and Kajiyama's Mechanical Models of the Human Vocal Cavity," *Journal of the Phonetic Society of Japan*, vol. 5, no. 2, pp. 31–38, 2001.

[12] K. W. Bozeman, "The Role of the First Formant in Training the Male Singing Voice," *Journal of Singing: The Official Journal of the National Association of Teachers of Singing*, vol. 66, no. 3, pp. 291–296, 2010.

[13] H. T. Kim, "Vocal feminization for transgender women: Current strategies and patient perspectives," *International Journal of General Medicine*, vol. 13, pp. 43–52, 2020.

[14] D. Kawitzky and T. McAllister, "The Effect of Formant Biofeedback on the Feminization of Voice in Transgender Women," *Journal of Voice*, vol. 34, no. 1, pp. 53–67, 2020. [Online]. Available: https://doi.org/10.1016/j.jvoice.2018.07.017

[15] J. Meister, H. Kühn, W. Shehata-Dieler, R. Hagen, and N. Kleinsasser, "Perceptual analysis of the male-to-female transgender voice after glottoplasty-the telephone test," *The Laryngoscope*, vol. 127, no. 4, pp. 875–881, apr 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/lary.26110

[16] J. Robbins, H. B. Fisher, and J. A. Logemann, "Acoustic characteristics of voice production after Staffieri's surgical reconstructive procedure," *Journal of Speech and Hearing Disorders*, vol. 47, no. 1, pp. 77–84, 1982.

[17] J. L. Kelly Jr. and C. C. Lochbaum, "Speech Synthesis," in *Fourth International Congress on Acoustics*. Copenhagen, Denmark: Organization Committee of the Fourth International Congress on Acoustics, 1962, pp. 838–840.

[18] M. Petyt, "Finite Element Techniques for Acoustics," in *Theoretical Acoustics and Numerical Techniques*, 1st ed. Vienna: Springer Vienna, 1983, ch. Finite Ele, pp. 51–103. [Online]. Available: http://link.springer.com/10.1007/978-3-7091-4340-7_2

[19] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics*, 4th ed. Wiley, 2000. [Online]. Available: https://www.wiley.com/en-gb/Fundamentals+of+Acoustics,+4th+Edition-p-9780471847892

[20] M. Kleiner, *Electroacoustical Analogies*, 1st ed. CRC Press, 2013, no. 1.

[21] J. L. Flanagan, *Speech Analysis Synthesis and Perception*, 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1972. [Online]. Available: http://link.springer.com/10.1007/978-3-662-01562-9

[22] K. Johnson, "The Acoustic Theory of Speech Production: Deriving Schwa," in *Acoustic and Auditory Phonetics*, 3rd ed. Wiley-Blackwell, 2012, pp. 23–40.

[23] D. Aalto, J. Malinen, and M. Vainio, "Formants," in *Oxford Research Encyclopedia of Linguistics*. Oxford University Press, jun 2018. [Online]. Available: http://linguistics.oxfordre.com/view/10.1093/acrefore/9780199384655.001.0001/acrefore-9780199384655-e-419

[24] M. Arnela, O. Guasch, and F. Alías, "Effects of head geometry simplifications on acoustic radiation of vowel sounds based on time-domain finite-element simulations," *The Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 2946–2954, 2013.

[25] W. J. Rugh, *Linear System Theory (2nd Edition)*, 2nd ed. Prentice Hall, 1995. [Online]. Available: http://www.amazon.com/Linear-System-Theory-2nd-Edition/dp/0134412052

[26] G. Fant, *Acoustic Theory of Speech Production: With Calculations based on X-Ray Studies of Russian Articulations.* Berlin, Boston: De Gruyter Mouton, 2012.

[27] G. Fant, J. Liljencrants, and Q. Lin, "A four-parameter model of glottal flow," *STL-Quarterly Progress and Status Report*, vol. 26, no. 4, pp. 1–13, 1985.

[28] A. E. Rosenberg, "Effect of Glottal Pulse Shape on the Quality of Nattural Vowels," *The Journal of the Acoustical Society of America*, vol. 46, no. 1A, pp. 82–82, 1969.

[29] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*, ser. Communication and Cybernetics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, vol. 12. [Online]. Available: http://link.springer.com/10.1007/978-3-642-66286-7

[30] D. H. Klatt, "Review of text-to-speech conversion for English," *The Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, sep 1987. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.395275

[31] A. W. Black and K. A. Lenzo, "FestVox: Building Synthetic Voices," 2014. [Online]. Available: http://festvox.org/bsv/

[32] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on*

Acoustics, Speech and Signal Processing. IEEE, may 2013, pp. 7962–7966. [Online]. Available: http://ieeexplore.ieee.org/document/6639215/https://ieeexplore.ieee.org/document/6639215/

[33] C. H. Shadle and R. I. Damper, "Prospects for Articulatory Synthesis: A Position Paper," in *Fourth ISCA Tutorial and Research Workshop on Speech Synthesis*, 2001, pp. 121–126.

[34] J. O. Smith III, "Physical Modeling Using Digital Waveguides," *Computer Music Journal*, vol. 16, no. 4, p. 74, 1992. [Online]. Available: https://www.jstor.org/stable/3680470?origin=crossref

[35] G. Molesini, "Geometrical Optics," in *Encyclopedia of Condensed Matter Physics*. Elsevier, 2005, pp. 257–267. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B0123694019006069

[36] A. S. Glassner, *An Introduction to Ray Tracing*. Academic Press Ltd., 1989.

[37] F. Pind, C.-h. Jeong, H. S. Llopis, K. Kosikowski, and J. Strømann-andersen, "Acoustic Virtual Reality – Methods and challenges," *Baltic-Nordic Acoustics Meeting*, no. April 2018, pp. 1–11, 2018.

[38] M. Vorländer, D. Schröder, S. Pelzer, and F. Wefers, "Virtual reality for architectural acoustics," *Journal of Building Performance Simulation*, vol. 8, no. 1, pp. 15–25, 2015.

[39] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," *The Journal of the Acoustical Society of America*, vol. 138, no. 2, pp. 708–730, 2015. [Online]. Available: http://dx.doi.org/10.1121/1.4926438

[40] Reuben Thomas, "Wayverb - Realistic, fast impulse response synthesis," 2017. [Online]. Available: https://reuk.github.io/wayverb/

[41] H. Kuttruff, *Room Acoustics*, 5th ed. CRC Press, 2009.

[42] H. Lee and B. H. Lee, "An efficient algorithm for the image model technique," *Applied Acoustics*, vol. 24, no. 2, pp. 87–115, 1988.

[43] S. Siltanen, T. Lokki, and L. Savioja, "Rays or Waves? Understanding the Strengths and Weaknesses of Computational Room Acoustics Modeling Techniques," *Proceedings of the International Symposium on Room Acoustics, ISRA 2010 29–31*, no. August, pp. 1–6, 2010.

[44] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[45] A. Kulowski, "Algorithmic representation of the ray tracing technique," *Applied Acoustics*, vol. 18, no. 6, pp. 449–469, 1985.

[46] H. Kim, L. Hernaggi, P. J. Jackson, and A. Hilton, "Immersive spatial audio reproduction for VR/AR using room acoustic modelling from 360° images," *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, pp. 120–126, 2019.

[47] B. Miga and B. Ziołko, "Real-Time Acoustic Phenomena Modelling for Computer Games Audio Engine," *Archives of Acoustics*, vol. 40, no. 2, pp. 205–211, 2015.

[48] P. S. Heckbert and P. Hanrahan, "Beam Tracing Polygonal Objects." *Computer Graphics (ACM)*, vol. 18, no. 3, pp. 119–127, 1984.

[49] T. Funkhouser, P. Min, and I. Carlbom, "Real-time acoustic modeling for distributed virtual environments," *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*, pp. 365–374, 1999.

[50] P. Min and T. Funkhouser, "Priority-driven acoustic modeling for virtual environments," *Computer Graphics Forum*, vol. 19, no. 3, pp. 179–188, 2000.

[51] S. Siltanen, T. Lokki, S. Tervo, and L. Savioja, "Modeling incoherent reflections from rough room surfaces with image sources," *The Journal of the Acoustical Society of America*, vol. 131, no. 6, pp. 4606–4614, 2012.

[52] S. Laine, S. Siltanen, T. Lokki, and L. Savioja, "Accelerated beam tracing algorithm," *Applied Acoustics*, vol. 70, no. 1, pp. 172–181, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.apacoust.2007.11.011

[53] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *J. Acoust. Soc. Am.*, vol. 98, no. 6, pp. 3302–3308, 1995.

[54] J. Van Mourik and D. Murphy, "Explicit higher-order FDTD schemes for 3D room acoustic simulation," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 12, pp. 2003–2011, 2014.

[55] B. Hamilton and S. Bilbao, "FDTD Methods for 3-D Room Acoustics Simulation with High-Order Accuracy in Space and Time," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 11, pp. 2112–2124, 2017.

[56] D. Murphy, A. Kelloniemi, J. Mullen, and S. Shelley, "Acoustic Modeling Using the Digital Waveguide Mesh," *IEEE SIGNAL PROCESSING MAGAZINE*, vol. 31, pp. 55–66, 2007.

[57] M. Karjalainen and C. Erkut, "Digital waveguides versus finite difference structures: Equivalence and mixed modeling," *Eurasip Journal on Applied Signal Processing*, vol. 2004, no. 7, pp. 978–989, 2004.

[58] M. Hornikx, R. Waxler, and J. Forssén, "The extended Fourier pseudospectral time-domain method for atmospheric sound propagation," *The Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 1632–1646, 2010.

[59] L. Zheng and X. Zhang, "Numerical Methods," in *Modeling and Analysis of Modern Fluid Problems*. Elsevier, 2017, pp. 361–455. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780128117538000086

[60] D. Ostashev, Vladimir E. and Wilson, D. Keith and Liu, Lanbo and Aldridge, David F. and Symons, Neill P. and Marlin, "Equations for finite-difference, time-domain simulation of sound propagation in moving inhomogeneous media and numerical implementation," *The Journal of the Acoustical Society of America*, vol. 117, no. 2, pp. 503–517, 2005.

[61] R. Blumrich and D. Haimann, "A linearized Eulerian sound propagation model for studies of complex meteorological effects," *The Journal of the Acoustical Society of America*, vol. 112, no. 2, pp. 446–455, 2002.

[62] E. Salomons; R. Blumrich; D. Heimann, "Eulerian Time-Domain Model for Sound Propagation over a Finite-Impedance Ground Surface. Comparison with Frequency-Domain Models," *ACUSTICA - acta acustica*, vol. 88, pp. 483–492, 2002.

[63] M. Hornikx, T. Krijnen, and L. Van Harten, "OpenPSTD: The open source pseudospectral time-domain method for acoustic propagation," *Computer Physics Communications*, vol. 203, pp. 298–308, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cpc.2016.02.029

[64] W. S. Hall, "Boundary element method," in *Lecture Notes in Applied and Computational Mechanics*, 1st ed. Springer Netherlands, 1994, vol. 27, pp. 61–83.

[65] S. Kirkup, "The boundary element method in acoustics: A survey," *Applied Sciences (Switzerland)*, vol. 9, no. 8, pp. 1–8, 2007.

[66] J. A. Hargreaves and T. J. Cox, "A transient boundary element method model of Schroeder diffuser scattering using well mouth impedance," *The Journal of the Acoustical Society of America*, vol. 124, no. 5, pp. 2942–2951, 2008.

[67] M. Abdullahi and S. O. Oyadiji, "Acoustic wave propagation in air-filled pipes using Finite Element Analysis," *Applied Sciences (Switzerland)*, vol. 8, no. 8, 2018.

[68] A. G. Antebas, F. D. Denia, A. M. Pedrosa, and F. J. Fuenmayor, "A finite element approach for the acoustic modeling of perforated dissipative mufflers with non-homogeneous properties," *Mathematical and Computer Modelling*, vol. 57, no. 7-8, pp. 1970–1978, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.mcm.2012.01.021

[69] R. Leveque, *Finite Volume Methods for Hyperbolic Problems*, 1st ed. Cambridge University Press, 2002.

[70] S. Bilbao, "Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 7, pp. 1524–1533, 2013.

[71] L. Savioja, "Real-time 3D finite-difference time-domain simulation of low- and mid-frequency room acoustics," *13th International Conference on Digital Audio Effects, DAFx 2010 Proceedings*, pp. 1–8, 2010.

[72] J. Lopez, "Some comments about graphic processing unit (GPU) architectures applied to finite-difference time-domain (FDTD) room acoustics simulation Present and future trends," in *International Congress on Acoustics 2013*, 2013, pp. 1–9.

[73] S. Oxnard and D. Murphy, "Room impulse response synthesis based on a 2D multi-plane FDTD hybrid acoustic model," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, vol. 21, no. 9, pp. 1940–1952, 2013.

[74] M. Aretz and M. Vorländer, "Combined wave and ray based room acoustic simulations of audio systems in car passenger compartments, Part II: Comparison of simulations and measurements," *Applied Acoustics*, vol. 76, pp. 52–65, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.apacoust.2013.07.020

[75] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating interactive virtual acoustic environments," *AES: Journal of the Audio Engineering Society*, vol. 47, no. 9, pp. 675–704, 1999.

[76] N. Magnenat-Thalmann, H. Kim, A. Egges, and S. Garchery, "Believability and interaction in virtual worlds," *Proceedings of the 11th International Multimedia Modelling Conference, MMM 2005*, pp. 2–9, 2005.

[77] M. Slater and S. Wilbur, "A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments Presence governs aspects of autonomie responses and higher-level behaviors of a participant in a VE," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 6, pp. 603–616, 1997. [Online]. Available: https://www.mitpressjournals.org/doi/pdf/10.1162/pres.1997.6.6.603

[78] R. Mehra, L. Antani, S. Kim, and D. Manocha, "Source Directivity and Spatial Audio for Interactive Wave-Based Sound Propagation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 495–503, 2014.

[79] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, no. October, pp. 99–102, 2001.

[80] M. Pollow, K. V. Nguyen, O. Warusfel, T. Carpentier, M. Müller-Trapet, M. Vorländer, and M. Noisternig, "Calculation of head-related transfer functions for arbitrary field points using spherical harmonics decomposition," *Acta Acustica united with Acustica*, vol. 98, no. 1, pp. 72–82, 2012.

[81] B. Rafaely and A. Avni, "Interaural cross correlation in a sound field represented by spherical harmonics," *The Journal of the Acoustical Society of America*, vol. 127, no. 2, pp. 823–828, 2010.

[82] R. Mehra, A. Rungta, A. Golas, M. Lin, and D. Manocha, "WAVE: Interactive Wave-based Sound Propagation for Virtual Environments," *IEEE Trans-*

actions on Visualization and Computer Graphics, vol. 21, no. 4, pp. 434–442, 2015.

[83] R. A. Rathnayake and W. K. Wanniarachchi, "Image source method based acoustic simulation for 3-D room environment," *International Journal of Scientific and Technology Research*, vol. 8, no. 11, pp. 222–228, 2019.

[84] J. Sandvad, "Dynamic aspects of auditory virtual environments," *Journal of the Audio Engineering Society*, vol. 4226, 1996.

[85] G. D. Romigh, D. S. Brungart, R. M. Stern, and B. D. Simpson, "Efficient Real Spherical Harmonic Representation of Head-Related Transfer Functions," *IEEE Journal on Selected Topics in Signal Processing*, vol. 9, no. 5, pp. 921–930, 2015.

[86] R. Boucheron, "About acoustic field characteristics in the test section of a cavitation tunnel," *Ocean Engineering*, vol. 211, no. December 2019, p. 107616, 2020. [Online]. Available: https://doi.org/10.1016/j.oceaneng.2020.107616

[87] B. B. Monson, E. J. Hunter, A. J. Lotto, and B. H. Story, "The perceptual significance of high-frequency energy in the human voice," *Frontiers in Psychology*, vol. 5, no. JUN, pp. 1–12, 2014.

[88] M. Speed, D. Murphy, and D. Howard, "Modeling the vocal tract transfer function using a 3D digital waveguide mesh," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 2, pp. 453–464, 2014.

[89] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of Biomedical Optics*, vol. 15, no. 2, p. 021314, 2010.

[90] R. Testuz, Y. Schwartzburg, and M. Pauly, "Automatic Generation of Constructable Brick Sculptures," *Eurographics*, no. December, pp. 81–84, 2013. [Online]. Available: http://infoscience.epfl.ch/record/189856/files/AutomaticGenerationofConstructableBrickSculptures-Eurographics_1.pdf

[91] K. Kowalczyk and M. Van Walstijn, "Modelling frequency-dependent boundaries as digital impedance filters in FDTD and K-DWM room acoustics simulations," *Audio Engineering Society - 124th Audio Engineering Society Convention 2008*, vol. 2, no. May 2014, pp. 1108–1119, 2008.

[92] J. Schneider, "Understanding the Finite-Difference Time-Domain Method," 2010. [Online]. Available: www.eecs.wsu.edu/$\sim$schneidj/ufdtd

[93] J. Van Mourik, "Higher-order Finite Difference Time Domain Algorithms for Room Acoustic Modelling," Ph.D. dissertation, University of York, 2016. [Online]. Available: http://etheses.whiterose.ac.uk/15661/8/PhDThesis.pdf

[94] A. Westerdiep, "Online Voxeliser." [Online]. Available: drububu.com/miscellaneous/voxelizer

[95] M. Arnela, O. Guasch, S. Dabbaghchian, and O. Engwall, "Finite element generation of vowel sounds using dynamic complex three-dimensional vocal tracts," *ICSV 2016 - 23rd International Congress on Sound and Vibration: From Ancient to Modern Acoustics*, no. July, pp. 1–7, 2016.

[96] P. Šivancara and J. Horáček, "Numerical modelling of effect of tonsillectomy on production of Czech vowels /A/ and /I/," in *4th International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, MAVEBA 2005*, vol. 1, no. 1, 2005, pp. 4–7.

[97] Y. Wang, H. Wang, J. Wei, and J. Dang, "Mandarin vowel synthesis based on 2D and 3D vocal tract model by finite-difference time-domain method," *2012 Conference Handbook - Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2012*, vol. d, pp. 2–5, 2012.

[98] K. Kowalczyk and M. Van Walstijn, "Room acoustics simulation using 3-D compact explicit FDTD schemes," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 34–46, 2011.

[99] P. Boersma and D. Weenink, "Praat: doing phonetics by computer." [Online]. Available: https://www.fon.hum.uva.nl/praat/

[100] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, p. 297, 1965.

[101] W. M. Gentleman and G. Sande, "Fast Fourier Transforms - For Fun And Profit," in *Proceedings of the November 7-10, 1966, fall joint computer conference on XX - AFIPS '66 (Fall)*. New York, New York, USA: ACM Press, 1966, p. 563. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1464291.1464352

[102] K. J. Åström and R. M. Murray, *Feedback Systems - An Introduction for Scientists and Engineers*, 2nd ed. Princeton: Princeton University Press, 2008.

[103] T. Chiba and M. Kajiyama, *The vowel: Its nature and structure*. Tokyo: Tokyo-Kaiseikan, 1942.

[104] P. Boersma and D. Weenink, "Sound: To Formant (burg)." [Online]. Available: https://www.fon.hum.uva.nl/praat/manual/Sound__To_Formant__burg____.html

[105] J. P. Burg, "Maximum Entropy Spectral Analysis," Ph.D. dissertation, Stanford University, 1975.

[106] N. Andersen, "On the calculation of filter coefficients for maximum entropy spectral analysis," *GEOPHYSICS*, vol. 39, no. 1, pp. 69–72, feb 1974. [Online]. Available: https://library.seg.org/doi/10.1190/1.1440413

[107] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge: Cambridge University Press, 1992. [Online]. Available: https://www.grad.hr/nastava/gs/prg/NumericalRecipesinC.pdf

[108] J.-P. Berenger, "Three-Dimensional Perfectly Matched Layer for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, vol. 127, no. 2, pp. 363–379, sep 1996. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999196901813

[109] Q.-H. Liu and J. Tao, "The perfectly matched layer for acoustic waves in absorptive media," *The Journal of the Acoustical Society of America*, vol. 102, no. 4, pp. 2072–2082, oct 1997. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.419657

[110] H. Takemoto, P. Mokhtari, and T. Kitamura, "Acoustic analysis of the vocal tract during vowel production by finite-difference time-domain method," *The Journal of the Acoustical Society of America*, vol. 128, no. 6, pp. 3724–3738, 2010.

[111] D. M. Howard, "The Vocal Tract Organ: A New Musical Instrument Using 3-D Printed Vocal Tracts," *Journal of Voice*, vol. 32, no. 6, pp. 660–667, nov 2018. [Online]. Available: https://doi.org/10.1016/j.jvoice.2017.09.014https://linkinghub.elsevier.com/retrieve/pii/S0892199717303971

[112] A. J. Gully, H. Daffern, and D. T. Murphy, "Diphthong Synthesis Using the Dynamic 3D Digital Waveguide Mesh," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 2, pp. 243–255, 2018.

[113] Michael Dawson-Haggerty, "Trimesh," 2023. [Online]. Available: https://trimsh.org/

[114] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," *Proceedings of the 14th Python in Science Conference*, no. Scipy, pp. 18–24, 2015.

[115] Y. Hati, "LPC Torch," 2019. [Online]. Available: https://github.com/yliess86/LPCTorch

[116] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, no. 2018, pp. 1–15, 2018. [Online]. Available: https://doi.org/10.1016/j.wocn.2018.07.001

[117] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *The Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, mar 1980. [Online]. Available: https://asa.scitation.org/doi/10.1121/1.383940

[118] M. Feng and D. M. Howard, "The Dynamic Effect of the Valleculae on Singing Voice – An Exploratory Study Using 3D Printed Vocal Tracts," *Journal of Voice*, 2021. [Online]. Available: https://doi.org/10.1016/j.jvoice.2020.12.012

[119] T. Vampola, J. Horáček, and J. G. Švec, "Modeling the influence of piriform sinuses and valleculae on the vocal tract resonances and antiresonances," *Acta Acustica united with Acustica*, vol. 101, no. 3, pp. 594–602, 2015.

[120] Stratasys, "F123 Series 3D Printers," 2023. [Online]. Available: https://www.stratasys.com/uk/3d-printers/printer-catalog/fdm-printers/f123-series-printers

[121] DPA Microphones, "4060 Series Miniature Omnidirectional Microphone," 2023. [Online]. Available: https://www.dpamicrophones.com/lavalier/4060-series-miniature-omnidirectional-microphone

[122] K. Henkel, "Parabox STL," 2017. [Online]. Available: https://github.com/karl-/pb_Stl

[123] M. Puckette, "Pure Data," 2017. [Online]. Available: https://puredata.info/

[124] H. Hacihabiboğlu, B. Günel, and A. M. Kondoz, "Source directivity simulation in digital waveguide mesh-based room acoustics models," *Proceedings of the AES International Conference*, pp. 1–9, 2007.

[125] A. Kelloniemi, D. T. Murphy, L. Savioja, and V. Välimäki, "Boundary conditions in a multi-dimensional digital waveguide mesh," *ICASSP, IEEE Inter-*

national Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 4, no. June 2014, 2004.

[126] A. Kelloniemi, L. Savioja, and V. Välimäki, "Spatial filter-based absorbing boundary for the 2-D digital waveguide mesh," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 126–129, 2005.

[127] S. B. Shelley, "Diffuse boundary modelling in the digital waveguide mesh," Ph.D. dissertation, University of York, 2007.